

Final Report

Artificial Intelligence and Human Factors for
Multimedia Applications



Cardiff University
School of Computer Science and Informatics

Author:	Sin Ching Cheung - C1768184
Supervisor:	Dr. Hantao Liu
Moderator:	Professor Irena Spasic
Module Code:	CM3203
Module Name:	One Semester Individual Project
Credits:	40
Date:	15 th May, 2020

Contents

Abstract	v
Acknowledgement	vi
1 Introduction	1
1.1 Project Motivation	1
1.2 Project Aims and Objectives	1
2 Background	2
2.1 Eye Tracking	2
2.1.1 Pupil-Corneal Reflection (P-CR) Eye Tracking	2
2.1.2 Eye Tracking in Cognitive Linguistics	3
2.1.3 The Importance for Using Eye Tracker	3
2.2 Multi-Media in Relation to Eye Tracking	4
2.3 Trending Research on Predicting Human Eye Movement	4
2.3.1 Saliency Model	4
2.3.2 Human Saccade	5
3 Approach	6
3.1 Dataset	6
3.2 Data Processing	7
3.3 Visual Analysis	9
3.4 Histogram Analysis and Comparison	9
3.5 Joint Saccade Analysis	10
4 Implementation	11
4.1 Project Environment	11
4.2 Data Processing	12
4.2.1 Renaming Incorrect Files	12
4.2.2 Saccade Calculation and Grouping Data	12
4.3 Histogram Analysis	14
4.4 Joint Saccade Profile	18
4.5 Example of plotting function utilisation	18
5 Result and Evaluation	21
5.1 Category Analysis	21
5.1.1 Saccade Amplitude	21
5.1.2 Saccade Orientation	22
5.1.3 Joint Saccade	23
5.2 Distortion Type Analysis	23
5.2.1 Saccade Amplitude	24
5.2.2 Saccade Orientation	25
5.2.3 Joint Saccade	27
5.3 Distortion Level Analysis	27
5.3.1 Saccade Amplitude	27
5.3.2 Saccade Orientation	29
5.3.3 Joint Saccade	30

5.4 Result Summarisation	30
------------------------------------	----

List of Figures

1	Indication of the corneal reflection shown by the white dots at different gaze positions (Blignaut 2014)	2
2	Demonstration of the pupil light response shown in the diagram where data are plotted when gazed at the blue or red light on a computer display (N=10 trials per colour) (Obaidellah et al. 2018)	3
3	Example of polar plot with and without convolution	10
4	Saccade amplitude on different image categories	21
5	Visual comparison of saccade amplitude (Rayleigh)	22
6	Saccade orientation on different image categories	22
7	Visual comparison of saccade orientation (Rayleigh)	23
8	Joint saccade on different image categories	23
9	Saccade amplitude on different distortion types	24
10	Visual comparison of saccade amplitude (Rayleigh)	24
11	Histogram comparison of saccade amplitude on different distortion types	25
12	Saccade orientation on different distortion types	25
13	Visual comparison of saccade orientation (Rayleigh)	26
14	Histogram comparison of saccade orientation on different distortion types	26
15	Joint saccade on different distortion levels	27
16	Saccade amplitude on different distortion levels	27
17	Visual comparison of saccade amplitude (Rayleigh)	28
18	Histogram comparison of saccade amplitude on different distortion levels	28
19	Saccade orientation on different distortion levels	29
20	Visual comparison of saccade orientation (Rayleigh)	29
21	Histogram comparison of saccade orientation on different distortion levels	30
22	Joint saccade on different distortion types	30

List of Tables

1	Example of image fixation data	6
2	Overview of dataset images in different distortion types and levels	7
3	Example of image fixation data with calculated saccade	8
4	Project libraries requirement and introduction	11

Abstract

Eye tracking research has often been experimented with high resolution images in which rarely discussed about the effect on how human and machine treats a distorted image. The study of human saccade could help to improve visual quality of a visual scene and help to aid the development of an intelligent image and video compression algorithms. Some researchers have proven that when traditional saliency model takes human saccade tendency into account, this could potentially predict human scan path. In this project, an experiment was done to analyse human saccade on different types and levels of distorted images and this study performs a data visualisation analysis and histogram to conduct a comparison between the results. The experiment is managed through data analysis to study human saccade behaviour in distorted images, especially in analysing the viewing bias of different distortion types and different level of its strengths. The result has shown that the human saccade of the given dataset has a strong horizontal and short distance bias rather than long travel and vertical movement when gazing an image. Images in different types and levels of distortion have not vary a vast difference on the saccade tendency. In contrast, machine may react enormously into micro changes.

Acknowledgement

I would like to express my sincere appreciation to the people who have given me all the support throughout my study in this research project. I wish to express my thankfulness to my supervisor Dr Hantao Liu for his patience and enthusiasm during the whole project, future advice on what's the next stepping stone, helpful information towards the research and inspirational ideas that have helped me tremendously throughout my research and the completion of this final year project. It is his non stoppable attitude and knowledge, plentiful experience and professional insight in data science and eye tracking that has enabled me to complete this research project successfully. Without his support and guidance, I wouldn't have completed this work. I would also like to thank my parents and family members for the unconditional care and support during my study, it is their eternally encouragement to never give up and go all the way to the end that have me standing right here today. A special credit goes to one of my family members who has helped me all the way through in researching and the completion of this project, without her creational ideas and her time to help me through, I wouldn't have completed this work to my best ability.

1 Introduction

1.1 Project Motivation

Human eye movement allow us to perceive our surroundings, controlling our mind and enriching sources of information on what we see every day predicting aspect of our personality and characteristic (Hoppe et al. 2018). There are four different types of characteristics in the function of eye movement, these functions are known as: 1) saccades which is known to be its rapidity in the movement of the eye from two fixation point, it consist of two main component which is the saccade amplitude and saccade orientation that measures the distance and angle between two fixation point, respectively 2) smooth pursuit movements which is a low progress of tracking eye movement, observing the movement stimulus of the fovea 3) vergence movements involves the alignment of the fovea of each eye targeting different directions from the observer and 4) vestibulo-ocular movements compensate head movement ensuring that visual images does not slip from the movement of different position (Purves et al. 2001). This implies the importance of saccade in such research area of eye tracking bringing the beneficial studies on future human eye movement prediction application. Human saccade could be an inspirational addition knowledge of the human mind, that could also be the future trend of technology.

There are several studies that has pinpointed out eye tracking in which has already embedded in various industry in today's world of high expertise industry, these studies includes web design as well as gaming (Djamasbi et al. 2010), cognition (Grillon et al. 2006), and various industry has already utilised eye tracking technique and the study of eye movement is being employed in human computer interaction (HCI) (Goldberg et al. 2002, Jacob 1990). This will aid to develop smarter algorithm on the image and video compression as well as improving visual quality of a visual scene.

When comes to human eye prediction, most of the research are done with high quality resolution images where distorted images has drawn less attention in the field. (Rai 2019) has proven distorted image will affect the performance of the deep learning model from predicting human eye movement. This will be interesting to find out the difference between a machine perspective vs human perspective on handling distorted images. The predicting result of machine will be affected by the image quality but would distorted images affect the tendency of how human browse an image differently in comparison with high resolution images?

1.2 Project Aims and Objectives

The aim of the early stages of the research, I would like to convert the raw eye tracking dataset into meaningful information using data processing technique. In addition, creating informative and adequate representation of the data to have a general visualisation of the data shape for initial analysis. Moreover, compare and find the difference in sequences and characteristic of the saccade profile of images in Social and Object category as well as in different distorted types and levels. At the later stage of the project, I will utilise the experimented data to draw a conclusion on how overall a human behave on each distorted types and levels.

2 Background

2.1 Eye Tracking

The state of the human mind is very extraordinary to study. It concludes the concept in eye tracking which has been around since 1800s creating today's modern eye tracking technology, allowing us to gather the best perception of the human mind (Tobiiproh n.d.). It acknowledges us to gather a supreme amount of insight of what influences their decisions and actions and the opportunity to knowledge about how people gather information (Tobiiproh n.d.). The process of eye tracking involves the measurement of eye movements by which determines the spot of where a person is looking and how long their gaze is at – this is also known as the ‘point of gaze’ (Research n.d.). The learning of this technique is commonly used by researchers due to the precise measures and understanding of visual attention, it also helps to gain a better insight into people's choice of information for task performance. For example, researchers are increasingly captivated by the changes in pupillometry in eye tracking as it reveals understandings into cognitive linguistics (Mathôt 2018).

This straightforward technology emits an invisible near-infrared (NIR) light beam reflecting its light in the person's corneal. The software uses image processing algorithms to calculate the main location point of the images sent by the eye tracking camera which then determines where the eye gazed the image on, creating a collection of fixation data. The position of the eye can be mapped multiple times a second due to the possibility of fine eye movement and the study of visual behaviour (Tobiiproh n.d.).

2.1.1 Pupil-Corneal Reflection (P-CR) Eye Tracking

The pupil-corneal reflection (P-CR) technique is commonly advertised as an interactive application to interact and observe eye movement. This technique was first applied in the oculometer in 1967 for Merchant study (Merchant 1967), since then this method is commonly used in the majority of the video eye tracker as this is the only way to determine gaze position using pupil position in their camera image (Hooge et al. 2016). The figure below demonstrates what the camera sees when the eye rotates to different directions, however the center of the corneal reflection will remain in a fix position only showing the movement of the center of the pupil.

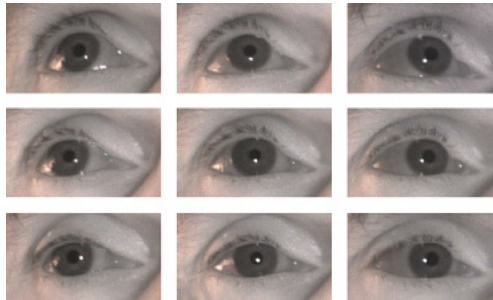


Figure 1: Indication of the corneal reflection shown by the white dots at different gaze positions (Blignaut 2014)

2.1.2 Eye Tracking in Cognitive Linguistics

Eye tracking has many beneficial in today's world of technology such as it could be used as a learning tool for researchers and businesses. Eye tracking is known to its popularity in research diagnosis for studying the development of cognitive process in neuroscience, it is therefore one of the main tools to investigate individual developmental trajectories (Hessels & Hooze 2019). An example of eye tracking in cognitive development is the study of pupillometry in pupil responses to light, near fixation and dilation in which is triggered by a stimulation in response to increase cognitive activity. Below is an example diagram profile of the pupil light reflex in response to blue and red light where the pupil is constricted and the pupil response to darkness where the pupil is dilated (Obaidellah et al. 2018).

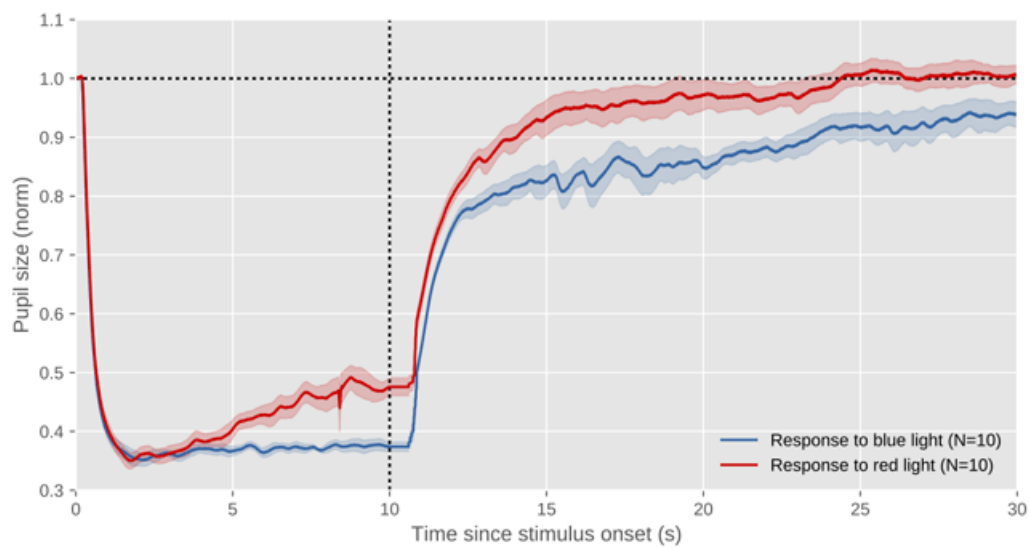


Figure 2: Demonstration of the pupil light response shown in the diagram where data are plotted when gazed at the blue or red light on a computer display (N=10 trials per colour) (Obaidellah et al. 2018)

2.1.3 The Importance for Using Eye Tracker

The study of eye tracking is very interesting to analyse, it concludes different usages such as online media and businesses where it could demonstrate the effectiveness and efficiency of advertisements. For example, advertiser can utilise the data analysis result from the eye tracker to determine the best position of an image for embedded advertisement. Data acquired by an eye tracker could be used by data analyst for further analysis to gain an insight of people's visual preferences and having an understanding of their state of mind in terms of what interest them the most within the image. Data analyst can also use this to determine the main object of people's awareness to create a more usability media that will attract more attention as a whole. For future preferences, eye tracker could potentially become an innovation interaction method to devises such as computer (Obaidellah et al. 2018) and it could also apply in the artificial intelligence (AI) industry which could lead to the ability to gain and achieve an stimulation of an human look, detecting the gestures from an human and provide richer insight into human emotion and this could be very

beneficial for future auto car industry.

2.2 Multi-Media in Relation to Eye Tracking

Digital technology is progressively used all over the world in the educational institution thereby eye tracking in multi-media has become one of the main areas of research, allowing a more fundamental and dynamic format in learning. Multi-media involves the combination of various media collective to create an instructional message such as using photographs, diagrams and videos. Evidence have shown that multi-media learning usage is far more effective than just plain text as not only it serves decorative purposes but also a deep learning curve (Was et al. 2016).

Eye tracking has played an important part into this role serving a valuable tool into studying and potential support in cognitive processes. Multi-media learning has drawn lots of attention into cognitive learning where it explores cognitive activities. Eye tracker used in the multi-media studies indicates where a person gazed during the integration of text and pictures, learners tend to show sufficient attention to most of the pictures rather than the predominant reading text representing that multi-media in learning is more actively used to draw people's attention and make them gain a more efficient understanding into the learning (Was et al. 2016). Furthermore, data collected in eye tracking can enhance better multi-media materials for learners and gain a more diverse learning (Alemdag & Cagiltay 2018).

2.3 Trending Research on Predicting Human Eye Movement

One of the most thought-provoking research topics in eye tracking is the ability to predict human eye movement. Researchers have tended to predict human eye movement in a way of saliency of an image. This particular technique requires user to generate a saliency map through the saliency area of an image such as colour and other elements that might attract human attention. These researches have been proven that it is successful in generating a saliency map that locates the most attractive area on an image through various deep learning models. It takes human saccade tendency into consideration, which could simulate the bias of human eye movement when browsing an image. However, this tendency has created a trend of development on variety of saliency model in the field of eye tracking across the internet.

However, saliency map has a lack of ability to predict human gaze path without external factor to aid the process. An interesting discovery by (Le Meur & Coutrot 2016, Le Meur et al. 2017) have shown that with the combination of a joint saccade probability model with other calculation conditional factors such as the memory state of a particular location. The usage of saccade amplitude and orientation data could effectively predict human scan path based on the Markov processing (Le Meur & Liu 2015).

2.3.1 Saliency Model

The model of visual saliency is used to generate a saliency map based on the attractiveness of each individual pixel within the scene, which is used to predict human eye

movement with the deep learning technique. Generally, the research on saliency modelling is influenced by the two different types of independent factors to predict human eye movement with saliency map, this is categorised as: 1) top-down approach which is based on focusing a particular task on a high-level, guiding the attention via control mechanism and 2) bottom-up approach which is a task independent component consisting of low-level attention (Zhu et al. 2014, Koehler et al. 2014).

Bottom-up saliency approach uses an interaction of low-level image features such as colour, intensity, orientation, shape, T-conjunctions, X-conjunctions etc at various different scales to identify distinctive image location (Mahdi & Qin 2017). It consists the participant in a free viewing condition to look at the image without a particular task in mind whereas in comparison the top-down saliency approach for saliency prediction is to focus on a particular task whilst viewing the image such as memory test, this identifies the participant interest using prior knowledge and high-level visual factors (Mahdi & Qin 2017).

Studies has been shown that the bottom-up saliency approach is far more effective and a better detection method than the top-down saliency approach, this is because it can provide useful information without using prior knowledge and understanding about the scene in which has achieved a greater performance in multi-media applications. In contrast, the top-down saliency approach does apply its expediency but mostly in visual cognitive processes with high-level factors directing eye movement. At first many people will draw its attention to the specific task and only attracted by certain regions that is standing out when gazing at the scene, this is usually in respect to the low-level features; colours, intensity, orientation or shape. However, this only happens during the first 1 minute then the eye gazed will focus immediately to the specific objects that is in the scene of the image such as pedestrians, this demonstrates that it is as efficient and effective utilising high-level of information to help visual saliency detection performance (Zhu et al. 2014).

2.3.2 Human Saccade

Visual attention involves the eye movement of observation and it is composed of fixations and saccades. Saccade defines to the changes in rapid eye movement that shifts from one fixation point to another, this is mainly used for gazing an object of interest (Termsarasab et al. 2015). Saccade is known be ballistic due to the predetermination of the initial point of the movement and the system cannot respond to any changes made in the position (Andrew Go Lee 2020). This movement is extremely useful in evaluation of diagnostic because when visual acuity is highest in the retina of the eye, it can bring focus on the fovea but it is also useful in eye tracking where it is used to observe and calculate the saccade amplitude and saccade orientation for further analysis (Termsarasab et al. 2015).

The control of eye movement consists of two decisions: 1) temporal decision which is the duration of fixation where it determines when the eye movement moves from one fixation to another and 2) spatial decision which is the amplitude of a saccade measuring the distance of a single eye movement between two fixation point (Nattkemper & Prinz 1987). The studying of fixation duration and saccade amplitude shows a positive correlation between the temporal and spatial decision system. However, evidences are very limited at this point because the two decisions are system that work independently (Kapoula et al. 1983).

3 Approach

In this section, a general overview would be given to discuss the approach that has been taken into account. There has been numerous challenges and obstacles that will be mentioned as well as the solution towards the goals. Algorithms and equation will also be listed out below.

3.1 Dataset

The dataset used in the project was provided by my supervisor Dr Hantao Liu. It comes with a total of 120 multimedia images categorised into two categories Object and Social, including the variation of different distortion types and levels from the original image. The object images were about static objects without human and animals involve such as a photo of computer desk with accessories. The social images shows photos about people mainly in a group such as photo of a group of people attending a conference. Each of the image has come with an individual CSV file that documented the fixation data collected by an eye-tracker from multiple participant also other general information, see table 1 for an example of the table structure.

Stimulus	Participant	Eye L/R	Event Start Time	Event End Time	Event Duration	Fixation Position X [px]	Fixation Position Y [px]
Ob1.CC.1.png	P01	Left	34.5	210.4	176	453	287.3
Ob1.CC.1.png	P01	Left	254.4	430.4	176	783.4	431.6
Ob1.CC.1.png	P01	Left	450.4	754.4	304	863.6	412.3
Ob1.CC.1.png	P11	Left	2.8	382.8	380	583.6	542
Ob1.CC.1.png	P11	Left	430.8	782.8	352	1051.1	463.3
Ob1.CC.1.png	P11	Left	826.8	1334.7	507.9	1427.2	367.5

Table 1: Example of image fixation data

Three different types of image distortion methods were applied to the original images including JPEG compression, motion blur and color contrast. Each distortion type has been sectioned into three different levels from 1 having the least affect to the image quality and 3 being the most distorted image. See table 2 to view sample of the dataset images including different distorted types and level.

Levels Types	Original Image	Level 1	Level 2	Level 3
Color Contrast				
Compression				
Motion Blur				

Table 2: Overview of dataset images in different distortion types and levels

3.2 Data Processing

Data processing is a crucial step to help get started with the research project to bring insight and understanding to the raw data. It helps to extract meaningful information and aid to discover the sequence, structure as well as distribution of the data from observation and analysis. Moreover, acquire additional knowledge from the data to gain a deeper understanding to it.

Human saccade is a smooth and continuously eye operation between more than one fixation point (Cassin et al. 1984), which is the main research area of this project. To get start with this, a calculation of saccade amplitude d and orientation ϕ were performed using the fixation point data in the dataset individually for each entry. Saccade amplitude referees to the angular distance of a successful movement of the eye between two fixation points in which could be calculated with the distance formula. On the other hand, saccade orientation referees to the angle of the eye movement which can express in the symbol ϕ . The fixation data can be understood as coordinated system, in which the saccade amplitude and orientation could be calculated based on the calculation theory of calculating coordinated distance and angle in degree.

The saccade amplitude consider as in the Euclidean space, based on the Pythagorean theorem, the distance d of two fixation points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ can be calculated using Distance formula (Britannica 2015) shown in below:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

In order to calculate the saccade orientation in an Euclidean space, a Tangent function can be applied. Given two fixation points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ to calculate the degree

ϕ , it is necessary to find the delta of two points as delta X d_x and Y d_y (see below).

$$\begin{aligned} d_x &= x_1 - x_2 \\ d_y &= y_1 - y_2 \end{aligned} \quad (2)$$

After calculated the delta, apply the Tangent function as the following to both delta and transit it as an Inverse Tangent function to calculate the ϕ .

$$\begin{aligned} \tan(\phi) &= d_x/d_y \\ \phi &= \tan^{-1}(d_x/d_y) \end{aligned} \quad (3)$$

A general approach to calculate the saccade amplitude and orientation would normally be having a program looping through each entry in the table, and for the amplitude (distance of the movement) and orientation (degree of the movement) of an entry n would be calculated with the current X Y coordinate (fixation point) (X_n, Y_n) with the previous entry's coordinate (X_{n-1}, Y_{n-1}) . However, as mentioned in section 3.1, multiple participants eye tracking data were included as continuous entries (see table 1), where each participant initial fixation point would be considered in the calculation as well, which leads to miscalculated data entry. For example, the last fixation entry of participant A should not be calculated with participant B initial fixation point (which is the next row of A), as that should not be consider as a saccade if the data is coming from a different person. Therefore, a algorithm was used (see algorithm 1) to prevent the program from calculating every entry in the table, a example of the end result after the calculation could see table 3.

Algorithm 1: Calculating the saccade data for each entry

```

currentPar ← Current entry's participant;
prevX ← Previous fixation point X;
prevY ← Previous fixation point Y;
foreach participant ∈ entries do
    if participant == currentPar then
        Calculate the saccade amplitude;
        Calculate the saccade orientation;
    else
        currentPar ← participant;
    end
    prevX ← Current entry's fixation point X;
    prevY ← Current entry's fixation point Y;
end

```

Stimulus	Participant	Eye L/R	Event Start Time	Event End Time	Event Duration	Fixation Position X [px]	Fixation Position Y [px]	Distance	Degree
Ob3.CC.1.png	P01	Left	53.8	173.8	120	993.4	243.4		
Ob3.CC.1.png	P01	Left	217.8	437.7	219.9	549.4	404.8	472.4256132	2.792930157
Ob3.CC.1.png	P01	Left	473.7	1581.7	1107.9	428.2	534.9	177.8073396	2.320793449
Ob3.CC.1.png	P02	Left	208	340	132	890.9	581.2		
Ob3.CC.1.png	P02	Left	644.3	788.3	144	987.1	470.7	146.5083274	-0.854470496
Ob3.CC.1.png	P02	Left	844.3	1124.3	280	396.8	417.6	592.6834737	-3.051879852

Table 3: Example of image fixation data with calculated saccade

3.3 Visual Analysis

Data visualisation is the early stage approach which provides a visual representation of the data which help familiarise the data structure and its treading in general. The purpose of data visualisation is to give an insight using graphs and charts to epitomise a clear communication of the findings. Visual analysis gives a more instant idea of the overall result as well as its effectiveness in unique features of the graph and chart, also it creates attractiveness in the whole picture. The power of data visualisation is that when data is presented graphically, people tend to understand more when communicated visually because our brains interpret through verbally (Few 2013). For example, line graph is used in everyday businesses and researches because it demonstrates exactly what you are interpreting and easier to understand, it can also be categorise into different sections to aid comparison of the result.

For an initial approach of data analysis, the comparison has divided into three main group individually for best result, based on the images distortion types, levels and its category. For each group of these data, we grouped all images that have the same attributes into a data frame, for example, all images with colour contrast were grouped and will have an direct comparison with the group of images with applied JPEG compression and motion blur. A histogram will be generated for each group, and normalise the histogram based on the data probability distribution. Then overlap the line curve of the normalised line to observe the difference.

3.4 Histogram Analysis and Comparison

Histogram is a statistical representation of the distribution of a set of numerical data showing the frequency of the data in class intervals of equal size, which basically divided a numbers of bins and counts the data appearance in each bin. It illustrates the probability of a given variable plotted as similar to a bar chart along the x and y axis. Histogram provide a great insight to the data distribution and density across bins, which allows to familiarise the overall data shape.

Histogram could also utilise as an compression tools, in other words, finding the similarity between the given histograms represented as a distance. In order to strengthen the analysis given the saccade amplitude and orientation, to further analyses the differences between different distortion types and levels other than visual observation, numerous distance metrics could help achieve the result. For this project, it has chosen two histogram distance measurement equation Euclidean Distance and Histogram Intersection for assess the similarity of the given histogram data from different distorted types and levels of images.

Euclidean distance is the measurement of a line between two points in the Euclidean space and it is represented in up to 3 dimensional space. The same concept could be applied to histogram which assume both histogram have the same bin size and within the same range, taking on bin at a time from both histogram in order to calculate the data population between those two bins and this process repeats to square sum the results

together to get an overall difference between the histograms.

$$D_{L2} = \sqrt{\sum_i (h_1(i) - h_2(i))^2} \quad (4)$$

Histogram Intersection is another histogram metrics to measure the similarity between two histograms. The general idea of it is to overlap two histogram together, and calculate the overall intersected area then return an positive number between 0 and 1 (Saabas 2016). Which in this case was used to strengthen the result from the Euclidean distance, in case the equation has certain conditional bias that might affect the result and judgement.

$$D_{\cap} = 1 - \frac{\sum_i (\min(h_1(i), h_2(i)))}{\min(|h_1(i)|, |h_2(i)|)} \quad (5)$$

3.5 Joint Saccade Analysis

Instead of having the saccade amplitude and orientation analysis in a separate manner, jointing both attributes together might lead to a different point of view through discovering the dataset in multiple aspects. It allows to obtain new knowledge and different perspective from the data, possibly in a two dimensional manner. As a result, a combination of polar plot and two-dimensional histogram was used to visual the jointed saccade.

The data was first passed through into a two-dimensional histogram function to produce the data density in each bin (bin size as 100) as a list. Then divided the polar plot into 100 bins as well given the edges of the polar distance as the minimum and maximum of the saccade amplitude, on the other hand, given the edges of polar degree as the minimum and maximum of the saccade orientation. After that, simply bind the two-dimensional histogram data into the polar plot. However, the visual results of the plot is unclear. As a solution, the data has to go through a convolution process using the Gaussian-2D-Kernel. See figure 3 for an example result with and without the convolution process.

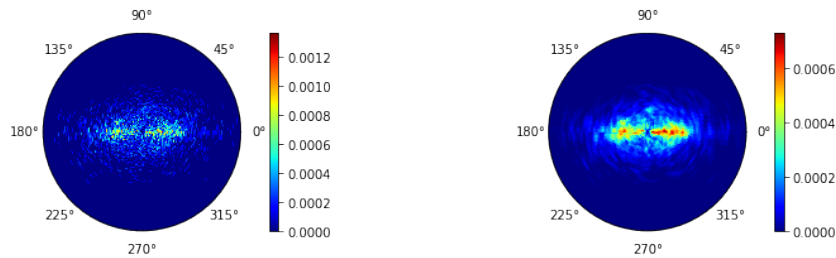


Figure 3: Example of polar plot with and without convolution

4 Implementation

4.1 Project Environment

The implementation part of this project was completed using the Python 3.7.6 along with Jupyter Notebook known as iPython, which is a powerful interactive scientific programming platform that provides live code preview. Allowing for instant visualise the result including graph plotting, helps researchers for effectively develop and testing code along the way. Its unique layout and operational mechanism as well as the rich functionality provides a much more convenient and effective environment that aid the research progress.

The following table describe the Python libraries that were used in the project and provide a brief introduction to each libraries.

Library	Description
os	This library provides the ability to Python to utilise the operating system functionality such as copying files.
re	This is an regex library for Python, which was used to match the string by using regular expression.
math	This library provide a variety of math function for programming such as sqrt root and other math theory.
matplotlib	Matplotlib is a plotting library which provide variety of different plots, allows user to visualise their numerical data in an human understandable way.
numpy	A great scientific computational package, mainly used it to handle array data.
pandas	This package provide a variety of class for handling data in a much convenient manner
scipy	Another scientific library, which implemented variety of scientific equation and algorithms.
itertools	This project utilised it to iterate and find all the combination of given values.

Table 4: Project libraries requirement and introduction

```
%matplotlib inline

import os
import re
import math
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.stats import gaussian_kde, rayleigh
from scipy.spatial.distance import euclidean
from itertools import combinations
```

```
from sklearn.metrics import mean_squared_error as mse
from astropy.convolution import convolve
from astropy.convolution.kernels import Gaussian2DKernel
```

4.2 Data Processing

4.2.1 Renaming Incorrect Files

As the JPEG compression file was originally found that has a naming error, where the level 1 and 3 compression level were mixed up. A simple Python script was created to rename the JPEG file. It simply loop through the directory contains the images and utilise regular expression to select the misnaming files and rename those files back to proper order.

```
import os
import re

def renameJPEG(path):
    cat = jpeg = num = ext = ''
    reSearch = ['JPEG_1', 'JPEG_3', 'JPEG_tmp']
    reName = ['tmp', '1', '3']
    for search, name in zip(reSearch, reName):
        for filename in os.listdir(path):
            if re.search(search, filename):
                if len(re.split('\W|_', filename)) == 4:
                    cat, jpeg, num, ext = re.split('\W|_', filename)
                else:
                    cat, jpeg, num, ext1, ext2 = re.split('\W|_', filename)
                    ext = ext1 + '.' + ext2
                newName = f'{path}{cat}-{jpeg}-{name}.{ext}'
                os.rename(f'{path}{filename}', newName)

if __name__ == '__main__':
    renameJPEG('./images/')
    renameJPEG('./data/')
```

4.2.2 Saccade Calculation and Grouping Data

The following function used for calculating the saccade amplitude given two fixation point with the Euclidean distance algorithm.

```
def calDistance(x1, y1, x2, y2):
    """
    This function calculates the distance between two pair of
    given coordinates using distance fomula.
```

```
Ref: https://www.purplemath.com/modules/distform.htm
"""
```

```
return math.sqrt(math.pow((x2 - x1), 2) + math.pow((y2 - y1), 2)
    ↪ )
```

As mentioned, the original eye tracking data were store in separate CSV file for every images, therefore, it is very time consuming and inconvenient to analyse directly. The listing below shows the process on how I gather the data from each csv file into a Panda dataframe with the give regex to identify which files will be grouped in together. The calculation for saccade amplitude and orientation are also done in this function.

```
def gatherDF(src, dest, regex, csv=True):
    """
    This function used to group all relevant dataset(csv) which
    have matches the regular expression and return a pandas dataframe.
    Also calculates the distance and degree between each moves.
    """
    df = pd.DataFrame()

    for filename in os.listdir(src):
        if re.search(regex, filename):
            df = df.append(pd.read_csv(f'{src}/{filename}'))

    df['Distance'] = np.nan
    df['Degree'] = np.nan
    currentPar = ''
    preX = float() # previous X coordinate of the participant
    preY = float() # previous Y coordinate of the participant

    df = df.reset_index()

    for index, row in df.iterrows():
        if row['Participant'] == currentPar:
            X = row['Fixation Position X [px]']
            Y = row['Fixation Position Y [px]']
            df.loc[index, 'Distance'] = calDistance(preX, preY, X, Y)
            df.loc[index, 'Degree'] = math.atan2(Y-preY, X-preX)
        else:
            currentPar = row['Participant']
            preX = row['Fixation Position X [px]']
            preY = row['Fixation Position Y [px]']

    if csv:
        df.to_csv(dest)
        print(f'saved to {dest}...')
    print(f'{dest} done.')
```

```
return df
```

This is an example on how to utilise the `gatherData()` function to group data using regular expression. Also shows the data I have been utilised and grouped during the project.

```
src = './data/'
dest = './grouped dataset/'

# !-- Original --!
originalImg = gatherDF(src, f'{dest}originalImg.csv', '[Soc|Ob]\d\.\
    ↪ png.csv', csv=True)

# !-- Object and Social --!
objectImg = gatherDF(src, f'{dest}objectImg.csv', '[Ob]\d\.\png.csv',
    ↪ csv=True)
socialImg = gatherDF(src, f'{dest}socialImg.csv', '[Soc]\d\.\png.csv'
    ↪ , csv=True)

# !-- Distortion types --!
colorContrast = gatherDF(src, f'{dest}colorContrast.csv', '_CC_', csv
    ↪ =True)
jpegCompression = gatherDF(src, f'{dest}jpegCompression.csv', '_JPEG_'
    ↪ , csv=True)
motionBlur = gatherDF(src, f'{dest}motionBlur.csv', '_MB_', csv=True)

# !-- Grouped by distortion levels --!
distLevel1 = gatherDF(src, f'{dest}distLevel1.csv', '_1\.(png|jpg)',
    ↪ csv=True)
distLevel2 = gatherDF(src, f'{dest}distLevel2.csv', '_2\.(png|jpg)',
    ↪ csv=True)
distLevel3 = gatherDF(src, f'{dest}distLevel3.csv', '_3\.(png|jpg)',
    ↪ csv=True)
```

4.3 Histogram Analysis

This is an function written for calculate the standard normal distribution.

```
def normalDist(bins, x):
    """
    This function calculates and return the standard normal
    distribution on a given histogram dataset.

    ref: https://en.wikipedia.org/wiki/Normal\_distribution
        (Standard normal distribution)
    """
```

```
mean = np.mean(x)
std = np.std(x)
return ((1 / (np.sqrt(2 * np.pi) * std)) * np.exp(-0.5 * (1 / std
    ↪ * (bins - mean)**2))
```

This function is used for plotting histogram. Which used matplotlib for histogram plotting with applied with different probability distortion function and kernel estimated function to get a normalised curve.

```
def plotHist(name, data, ax, title, saccade):
    """
    This funciton plot the histogram using the given dataset and
    apply three different distribution functions (Normal, Gaussian and
    Rayleigh) on it.
    """
    if saccade == 'amp':
        n, bins, patches = axs[ax].hist(data, binSize, histtype='
            ↪ stepfilled', density=True, range=(0, 1200))
        axs[ax].set_xlabel('Saccade Amplitude [px]')
    elif saccade == 'ori':
        n, bins, patches = axs[ax].hist(data, binSize, histtype='
            ↪ stepfilled', density=True, range=(-180, 180))
        axs[ax].set_xlabel('Saccade Orientation [degree]')

    # Standard Normal Distribution
    axs[ax].plot(bins, normalDist(bins, data), '--', label="Normal")

    # Gaussian Distribution
    gaussianDist = gaussian_kde(data)
    axs[ax].plot(bins, gaussianDist(bins), '--', label="Gaussian")

    # Rayleigh distribution
    param = rayleigh.fit(data)
    bestFit = rayleigh.pdf(bins, param[0], param[1])
    axs[ax].plot(bins, bestFit, '--', label="Rayleigh")

    # Axes configuration
    axs[ax].title.set_text(title)
    if ax == 0:
        axs[ax].set_ylabel('Probability')
    axs[ax].legend()

    return {
        "name": name,
        "data": data,
        "hist": [n, bins, patches],
        "normal": [bins, normalDist(bins, data)],
        "gaussian": [bins, gaussianDist(bins)],
```

```

    "rayleigh": [bins, bestFit]
}

```

This function plots a multi line graph in which overlap the result from the normalised histogram to provide a direct comparison on different histogram data.

```

def multiLinePlot(saType, title, *plotVal, normal=False, gaussian=
    ↪ False, rayleigh=False):
    """
    TODO: fill in comment
    """
    # Comparison of Saccade Amplitude
    fig, ax = plt.subplots(figsize=(15, 5))

    tmp = set()

    # Plotting
    for data in plotVal:
        if normal:
            ax.plot(data['normal'][0], data['normal'][1], label=f'{
                ↪ data["name"]} (Normal)')
            tmp.add('Normal')
        if gaussian:
            ax.plot(data['gaussian'][0], data['gaussian'][1], label=f'
                ↪ {data["name"]} (Gaussian)')
            tmp.add('Gaussian')
        if rayleigh:
            ax.plot(data['rayleigh'][0], data['rayleigh'][1], label=f'
                ↪ {data["name"]} (Rayleigh)')
            tmp.add('Rayleigh')

    # Axes configuration
    title = f'Visual comparison of saccade {saType} ({" + ".join(tmp)
        ↪ })'
    ax.set_title(title, fontsize=16)
    ax.set_xlabel('Saccade Amplitude [px]')
    ax.set_ylabel('Probability')
    ax.legend()

    plt.show()

```

This function used to calculate the histogram intersection given two histogram data and bin size.

```

def intersectDist(x1, x2, binSize):
    """
    This function compairs the similarity between two
    given histogram values using the intersection distance.

```

```
"""  
sm = 0  
for i in range(binSize):  
    sm += min(x1[i], x2[i])  
return sm
```

This is another function for plotting, which include calculation for given histogram data in which using Euclidean distance and Histogram Intersection to assess the similarity and difference between two given histogram.

```
def distanceMetricsPlot(saccade, histData):  
    # Plotting configuration  
    fig, axs = plt.subplots(1, 2, figsize=(15, 4), sharey=False,  
        ↪ tight_layout=True)  
    if saccade == 'amp':  
        fig.suptitle('Histogram comparison of saccade amplitude on  
            ↪ different distorted images', y=1.1, fontsize=16)  
    elif saccade == 'ori':  
        fig.suptitle('Histogram comparison of saccade orientation on  
            ↪ different distorted images', y=1.1, fontsize=16)  
    axs[0].set_title('Euclidean Distance')  
    axs[1].set_title('Histogram Intersection')  
    axs[0].set_ylabel('Distance')  
  
    # Calculation of both histogram distance metrics  
    combDisTypes = list(combinations(histData, 2))  
    xLabel = []  
    euc = []  
    inter = []  
    for sa in combDisTypes:  
        xLabel.append(f'{sa[0]["name"]}\n vs\n {sa[1]["name"]}\n')  
        eucDist = euclidean(sa[0]['hist'][0], sa[1]['hist'][0])  
        interDist = intersectDist(sa[0]['hist'][0], sa[1]['hist'][0],  
            ↪ binSize)  
        euc.append(eucDist)  
        inter.append(interDist)  
  
    axs[0].bar(xLabel, euc)  
    axs[1].bar(xLabel, inter)  
  
    # bar configuration  
    axs[0].grid(linestyle='--')  
    axs[1].grid(linestyle='--')  
  
    plt.show()
```


4.4 Joint Saccade Profile

This is the function that produce a polar plot to display the joint saccade visualisation. It first takes the minimum and maximum value of the saccade distance and degree on the given dataset, and used it to form the edges of the bins. Then put the distance and degree to a two-dimensional enthusiasm function to produce a 2d-histogram. And lastly plot the 2d-histogram on a polar plot with convolved by a Gaussian 2D kernel and produce the polar plot.

```
# Joint saccade polar plot function
def jointSaccadePolar(name, data):
    total = len(data)
    fig, axs = plt.subplots(1, total, figsize=(20, 6), sharey=False,
        ↪ tight_layout=True, subplot_kw={'projection': 'polar'})
    i = 0

    for n, d in zip(name, data):
        axs[i].set_title(n, y=1, fontsize=20)
        degree = d[d.Degree.notnull()]['Degree'].to_numpy()
        distance = d[d.Distance.notnull()]['Distance'].to_numpy()
        xedge = np.linspace(min(degree), max(degree), 100)
        yedge = np.linspace(min(distance), max(distance), 100)
        x, y = np.meshgrid(xedge, yedge)
        H, xedges, yedges = np.histogram2d(distance, degree, bins
            ↪ =100, density=True)
        p = axs[i].pcolormesh(x, y, convolve(H, Gaussian2DKernel(
            ↪ x_stddev=1)), cmap=plt.cm.jet)
        fig.colorbar(p, ax=axs[i])
        i += 1

    plt.show()
```

4.5 Example of plotting function utilisation

The listing below uses distortion types analysis as an example on demonstrating how the various function listed above was used in practice in order to produce different plotting to visualise the eye-tracking data.

```
# Distortion types analysis plotting example

binSize = 100

# Saccade amplitude on different distortion types
fig, axs = plt.subplots(1, 3, figsize=(15, 3), sharey=True,
    ↪ tight_layout=True)

# Figure configuration
```

```
fig.suptitle('Saccade amplitude on different distortion types', y
    ↪ =1.1, fontsize=16)

# !— Saccade Amplitude of Color Contrast —!
colorContrastDist = colorContrast[colorContrast.Distance.notnull()][
    ↪ 'Distance']
colorContrastData = plotHist('Color Contrast', colorContrastDist, 0,
    ↪ 'Color Contrast', 'amp')

# !— Saccade Amplitude of JPEG Compression —!
jpegCompressionDist = jpegCompression[jpegCompression.Distance.
    ↪ notnull()]['Distance']
jpegCompressionData = plotHist('JPEG Compression',
    ↪ jpegCompressionDist, 1, 'JPEG Compression', 'amp')

# !— Saccade Amplitude of Motion Blur —!
motionBlurDist = motionBlur[motionBlur.Distance.notnull()]['Distance
    ↪ ']
motionBlurData = plotHist('Motion Blur', motionBlurDist, 2, 'Motion
    ↪ Blur', 'amp')

# !— Multiline plot of Saccade Amplitude of all distortion types —!
multiLinePlot('amplitude', 'Object and Social', colorContrastData,
    ↪ jpegCompressionData, motionBlurData, gaussian=False, normal=
    ↪ False, rayleigh=True)

# Distance metrics (saccade amplitude)
histData = [colorContrastData, jpegCompressionData, motionBlurData]
distanceMetricsPlot('amp', histData)

# Saccade orientation on different distortion types
fig, axs = plt.subplots(1, 3, figsize=(15, 3), sharey=True,
    ↪ tight_layout=True)

# Figure configuration
fig.suptitle('Saccade orientation on different distortion types', y
    ↪ =1.1, fontsize=16)

# !— Saccade Orientation of Color Contrast —!
colorContrastDeg = colorContrast[colorContrast.Degree.notnull()]['
    ↪ Degree'] * 57.295779513
colorContrastData = plotHist('Color Contrast', colorContrastDeg, 0,
    ↪ 'Color Contrast', 'ori')

# !— Saccade Orientation of JPEG Compression —!
jpegCompressionDeg = jpegCompression[jpegCompression.Degree.notnull]
```

```
    ↪ ()['Degree'] * 57.295779513
jpegCompressionData = plotHist('JPEG Compression',
    ↪ jpegCompressionDeg, 1, 'JPEG Compression', 'ori')

# !— Saccade Orientation of Motion Blur —!
motionBlurDeg = motionBlur[motionBlur.Degree.notnull()]['Degree'] *
    ↪ 57.295779513
motionBlurData = plotHist('Motion Blur', motionBlurDeg, 2, 'Motion
    ↪ Blur', 'ori')

# !— Multiline plot of Saccade Orientation of all distortion types —!
multiLinePlot('orientation', 'Object and Social', colorContrastData,
    ↪ jpegCompressionData, motionBlurData, gaussian=True, normal=
    ↪ False, rayleigh=False)

# Distance metrics (saccade orientation)
histData = [colorContrastData, jpegCompressionData, motionBlurData]
distanceMetricsPlot('ori', histData)

# Distortion types joint saccade
jointSaccadePolar(["Color Contrast", "JPEG Compression", "Motion
    ↪ Blur"], [colorContrast, jpegCompression, motionBlur])

plt.show()
```

5 Result and Evaluation

From the results undertaken, all of the results have reflected to be significantly similar to each other, therefore, it makes it difficult to interpret each result and distinguish each one by one as most of them will be repetitively the same. From each plot there will be a brief description of the result, but a full conclusion will be drawn at the end to conclude my overall findings and a more in-depth explanation to analyse my results (see section 5.4).

5.1 Category Analysis

5.1.1 Saccade Amplitude

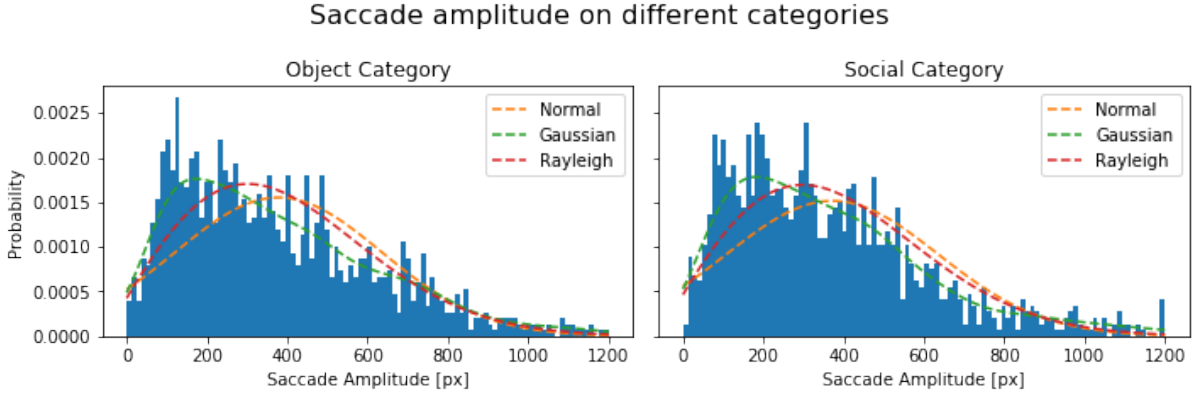


Figure 4: Saccade amplitude on different image categories

The above figure shows the saccade amplitude histogram on Object and Social images. The overall shape of these two histogram are not exactly the same, but commonly they both form a similar trend where the highest data density was around 100px to 300px and decrease accordingly after that until 800px after and started to smoothen. When plotting the normalised distortion of both histogram (see 5), it could easily see the two lines were also intersecting each other, which means the participants saccade amplitude tendency were indeed similar to each other.

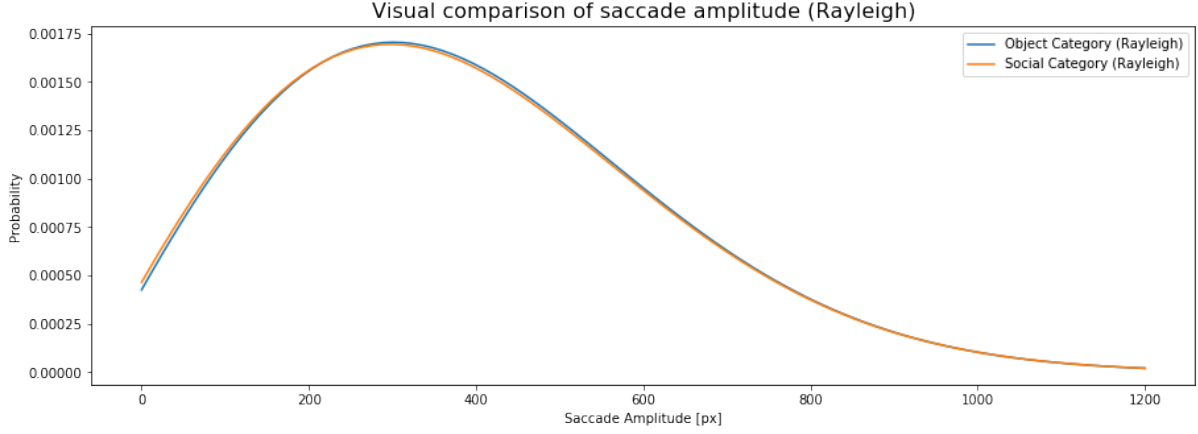


Figure 5: Visual comparison of saccade amplitude (Rayleigh)

5.1.2 Saccade Orientation

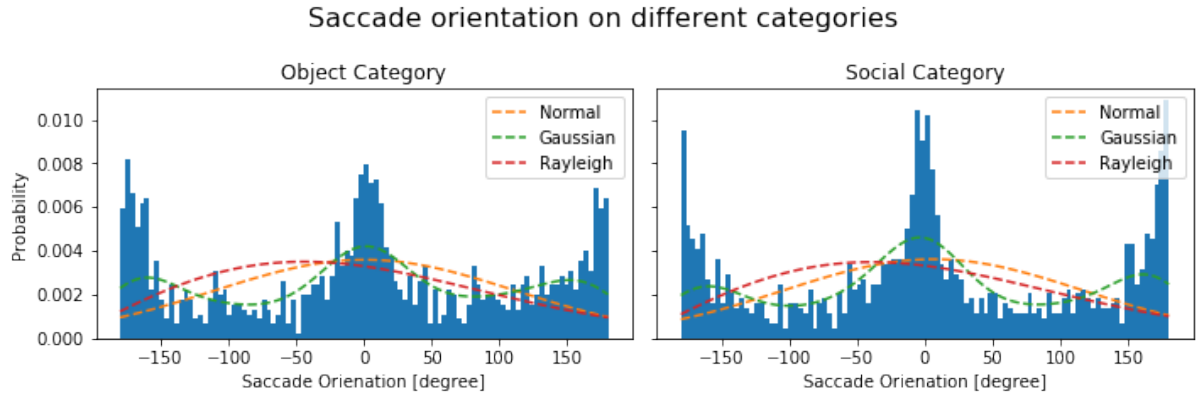


Figure 6: Saccade orientation on different image categories

The above figure shows the individual saccade orientation histogram for Object and Social images. Generally, both of them appear as a similar shape where the data density peaks were located at around -180° , 0° and 180° . In order to compare both histogram data directly, we have normalised both histogram with the Gaussian kernel estimated function and plot the result into a multi line chart. The graph has shown that Social images have a lower data density on the -180° than Object, but in the 0° and 180° have a higher data density.

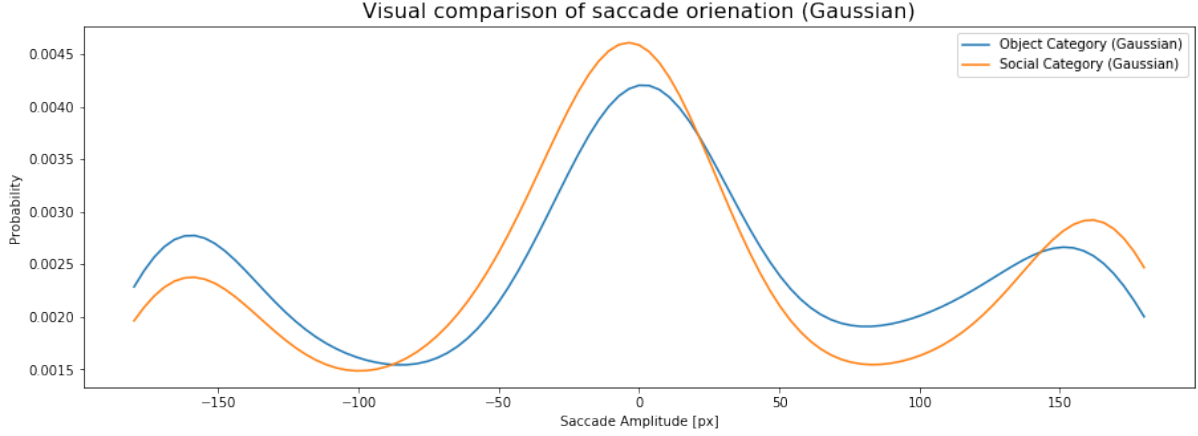


Figure 7: Visual comparison of saccade orientation (Rayleigh)

5.1.3 Joint Saccade

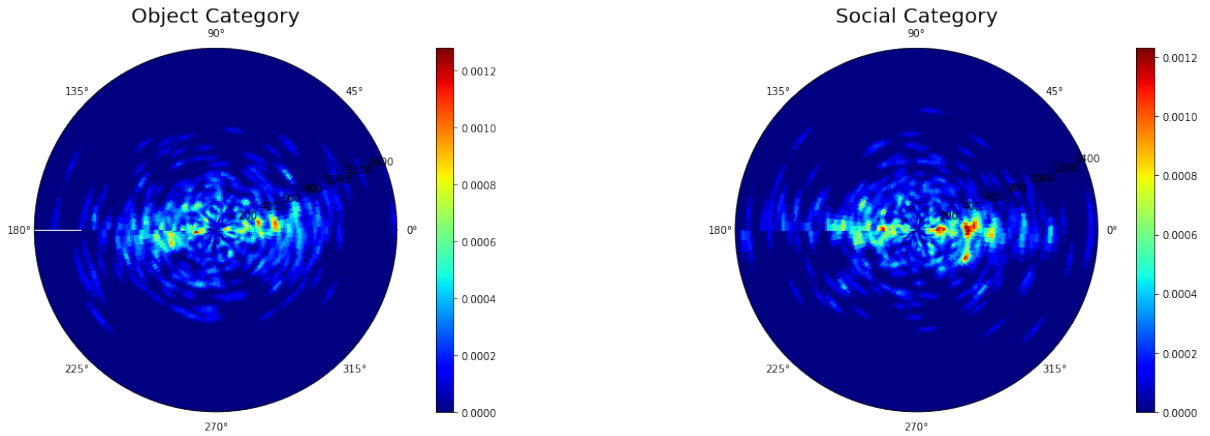


Figure 8: Joint saccade on different image categories

The above joint saccade polar plot is a bit hard to tell the overall tendency and visualise clearly of the distribution of the data due the quantity of the sample being too small. But if we look closer, it still can indicate that the horizontal short eye movement were still being the majority, it has reflected the human saccade tendency may not vary a significant amount from Object and Social category.

5.2 Distortion Type Analysis

The following section will analyse the human eye movement tendency while viewing different types of distorted images including colour contrast, JPEG compression and motion blur.

5.2.1 Saccade Amplitude

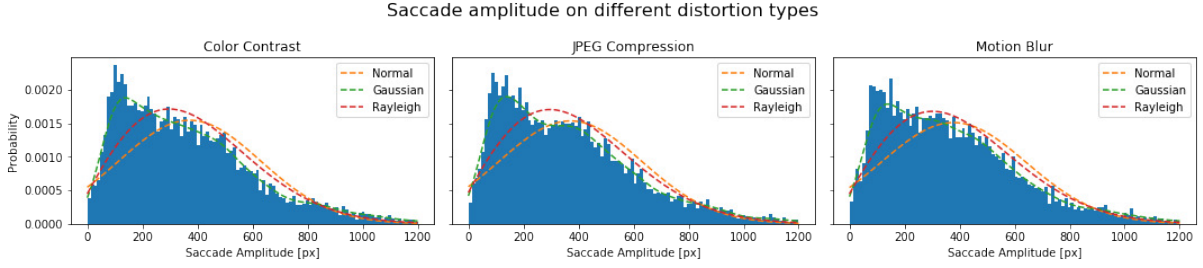


Figure 9: Saccade amplitude on different distortion types

Figure 9 shows the data distribution in density for each distortion type. All three of them have shown a very similar shape, where the participants prefer moving in such a short distance at around 100px - 200px. Which also shows an obvious data trend where the distance increases, the data density has decreased accordingly and smoothen after reaches 800px. That finding has reflected the participants are more likely to move in short distance rather than a long saccade.

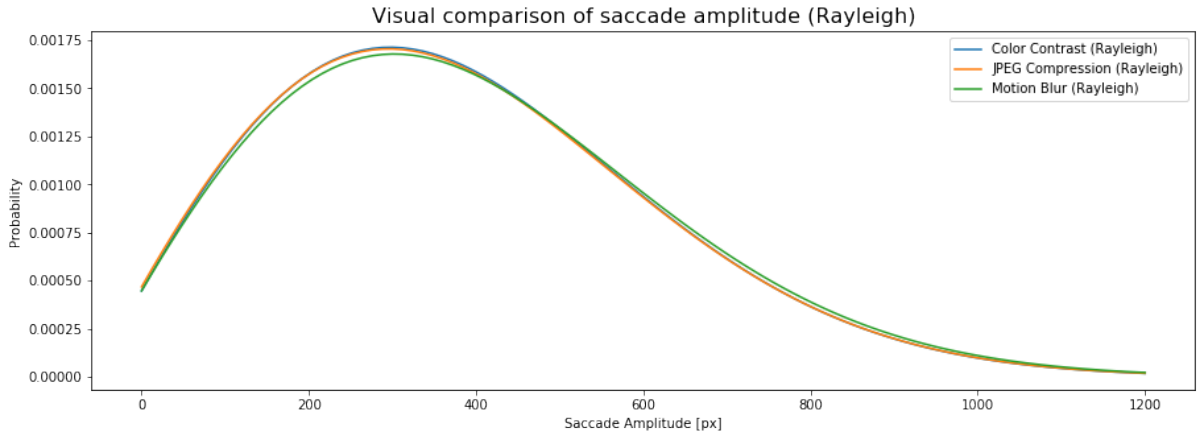


Figure 10: Visual comparison of saccade amplitude (Rayleigh)

The above figure overlaps the Rayleigh normalised histogram data from figure 9 for a direct comparison on those distortion types. The result has shown even the distortion effect were vary, the participants still having a really similar viewing preference with others. The result has reflected human viewing bias will not be affected by colour contrast, JPEG compression and motion blur.

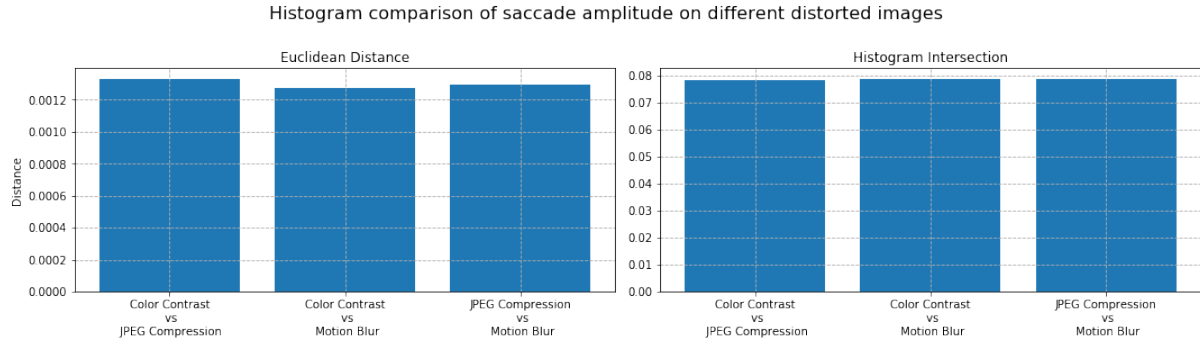


Figure 11: Histogram comparison of saccade amplitude on different distortion types

Figure 11 plotted the Euclidean Distance and Histogram Intersection in a bar chart for a visualisation, where the value of Euclidean Distance represent how distinguish given histograms are and the Histogram Intersection represent how similar given histograms are. Both histogram distance metrics have shown that the histogram result of different distortion types images in saccade amplitude has a very similar result in overall similarity.

5.2.2 Saccade Orientation

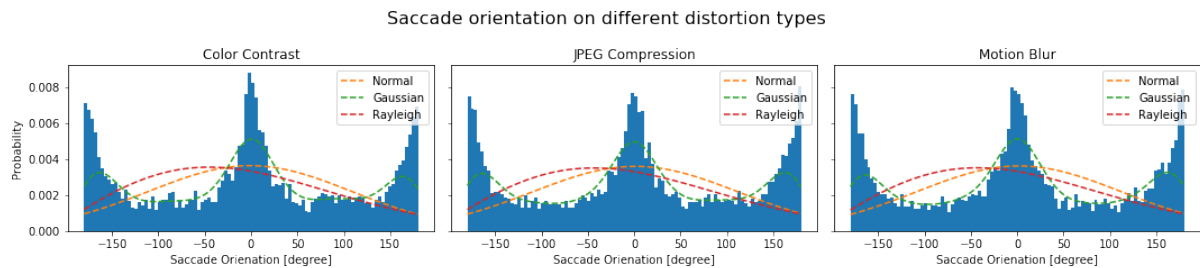


Figure 12: Saccade orientation on different distortion types

Above has shown the histogram plot of saccade orientation for each distortion types. The overall shape of the data were similar, which can clearly see that the participant has a strong bias on their eye moving angle in mostly in an horizontal manner.

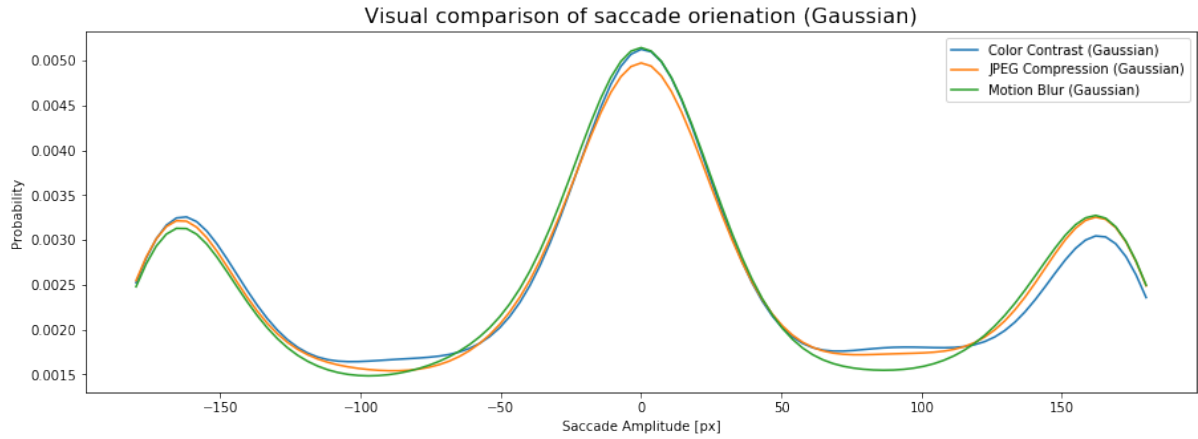


Figure 13: Visual comparison of saccade orientation (Rayleigh)

As same as comparing saccade amplitude, the above figure overlap the probability distribution result in comparison, but this time with normalised with an kernel estimated function as the appearing result were too random. The result has shown a strong similarity as well on the saccade angle tendency.

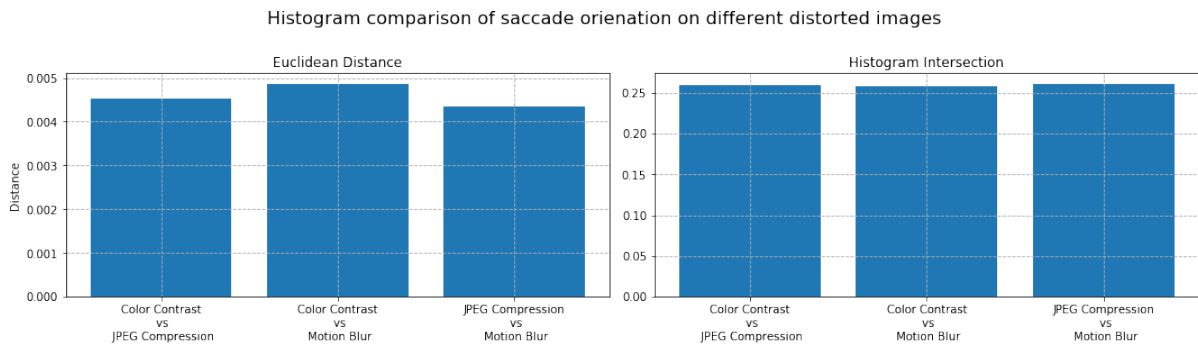


Figure 14: Histogram comparison of saccade orientation on different distortion types

The histogram result were once again strengthen the statement of the finding from the probability distortion analysis. Which also reflecting the overall sacacade orientation probability has remain in a really similar manner.

5.2.3 Joint Saccade

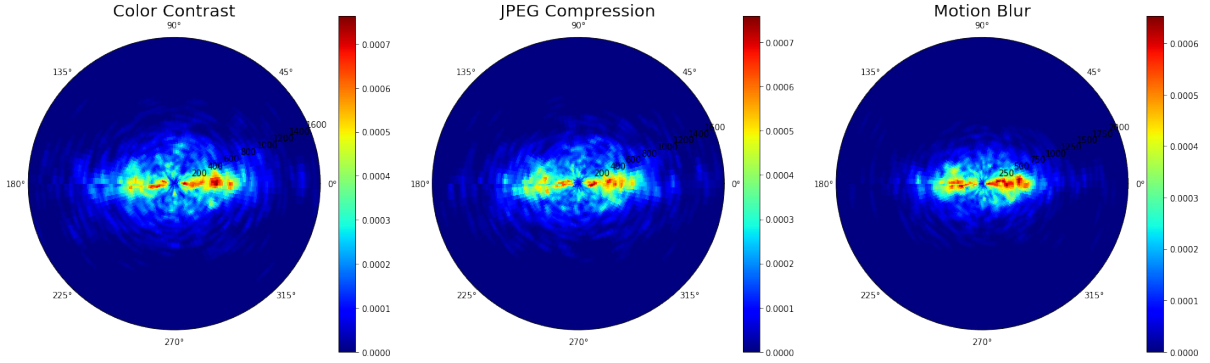


Figure 15: Joint saccade on different distortion levels

The above polar plot shows visualise the jointed saccade amplitude and orientation based on a two-dimensional histogram result. With the reddest represent the most dense area. Where the radius represent the saccade amplitude and the degree represent the saccade orientation. According to the figure, all three of distortion types eye movement have shown a strong central with short and horizontal movement bias.

5.3 Distortion Level Analysis

5.3.1 Saccade Amplitude

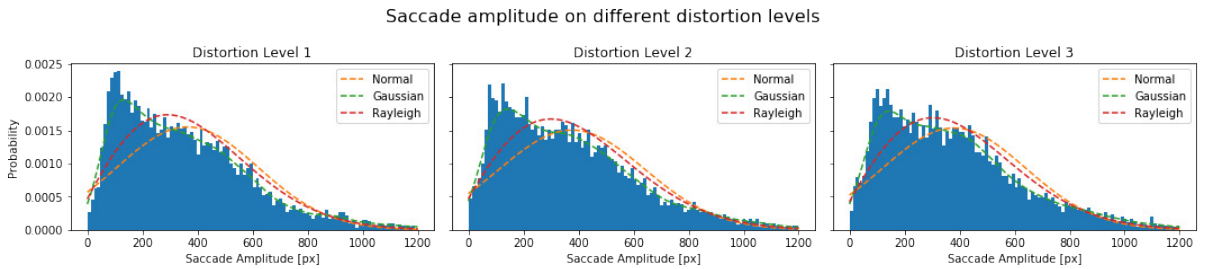


Figure 16: Saccade amplitude on different distortion levels

This figure indicates the overall peak from the three histogram results, it shows the participant intended to gaze at a short distance showing their human eye behaviour. The result from level 1 distortion have an overall one peak density at 0.0024 whereas level 2 and 3 distortion have two peak waves between 100-500 pixel of saccade amplitude.

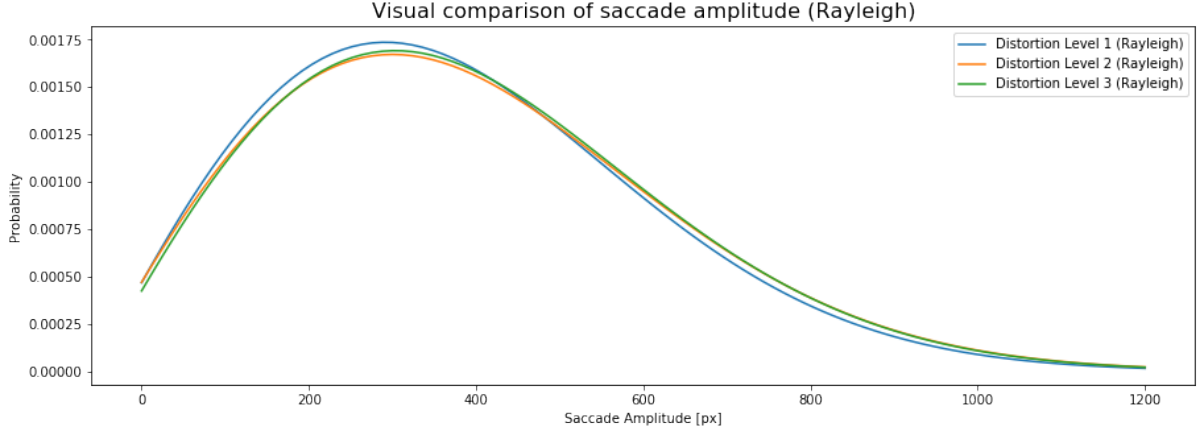


Figure 17: Visual comparison of saccade amplitude (Rayleigh)

Although figure 16 shows that the data have various peak waves, the result from figure 17 and 18 have shown that the data distribution still have the same similarities. From figure 17 shows three of those line are completely intersect with each other, only level 1 distortion have a slight higher peak level than the rest around 200-400 pixel. Results of histogram intersection from 18 has return a really close value meaning that based on the comparison, all three groups have a similar amount of intersection overlay on each others histogram. Moreover, the Euclidean distance has pointed out level 2 distortion are much distinguish from level 3 and level 1 and 3 have a rather less difference between them.

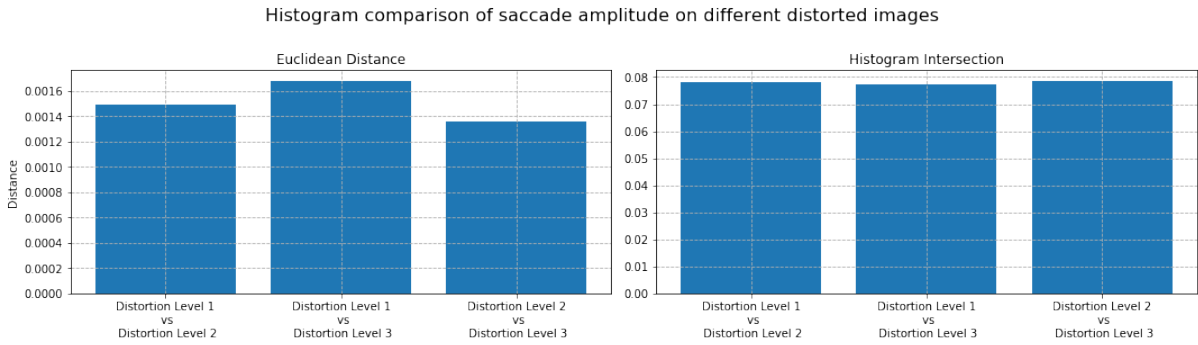


Figure 18: Histogram comparison of saccade amplitude on different distortion levels

5.3.2 Saccade Orientation

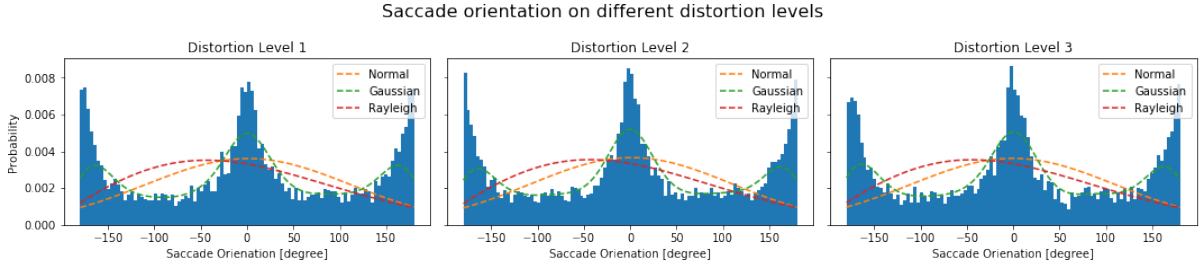


Figure 19: Saccade orientation on different distortion levels

In this figure, it illustrates an overflow of the peak level going from up to down of the distortion levels in saccade orientation. This is resulted by the degree angle between 180° , 0° and -180° , it shows how participant gaze between the different angles shown. For example, this figure demonstrates that the participant gazed the image more on where it is shown as the highest peaks and the low peaks indicates that the participant is less likely to gaze in that particular area angle in the scene.

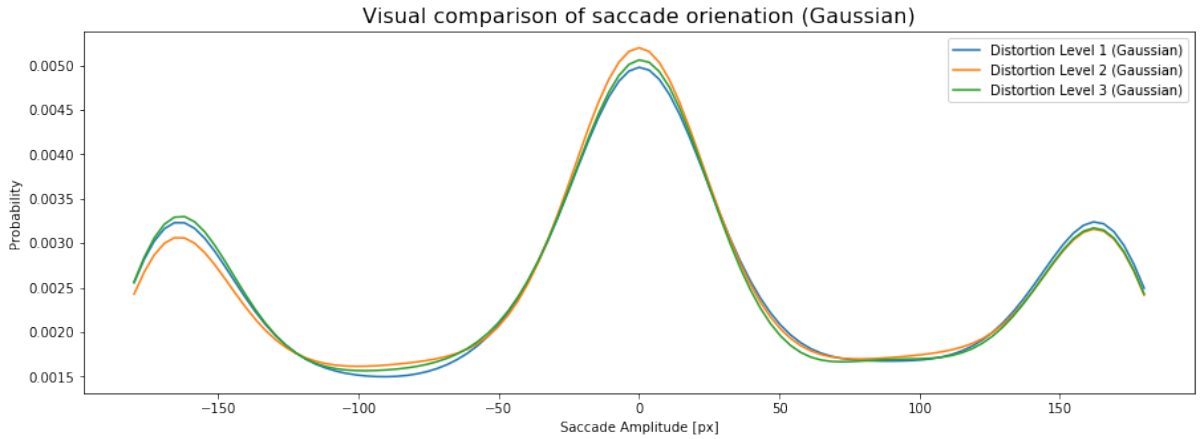


Figure 20: Visual comparison of saccade orientation (Rayleigh)

Figure 20 and 21 of the line and bar graph done to assess the similarity and distinguish between the comparison of saccade orientation at different distortion levels. The results demonstrates the visual comparison at level 1-3 distortion, showing that the level streak are very close in line of each other. As you can see in the Euclidean distance of the histogram shows how the bar are much as the same level, although between 1 and 3 distortion level is slightly higher.

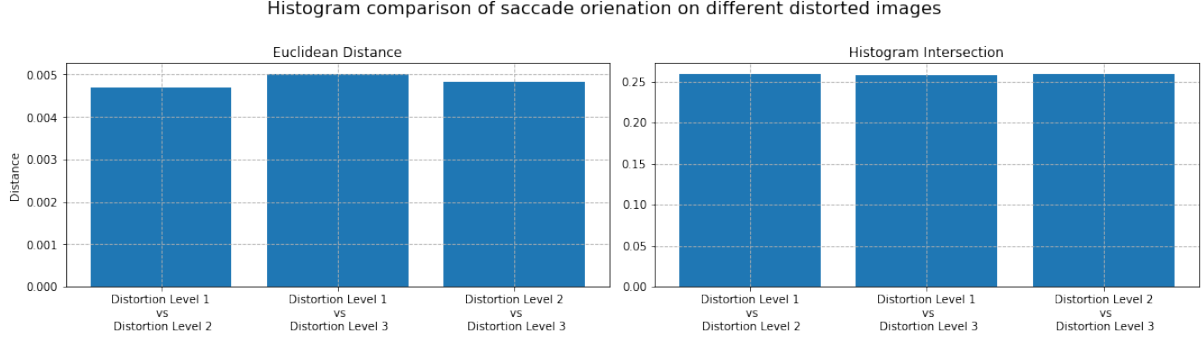


Figure 21: Histogram comparison of saccade orientation on different distortion levels

5.3.3 Joint Saccade

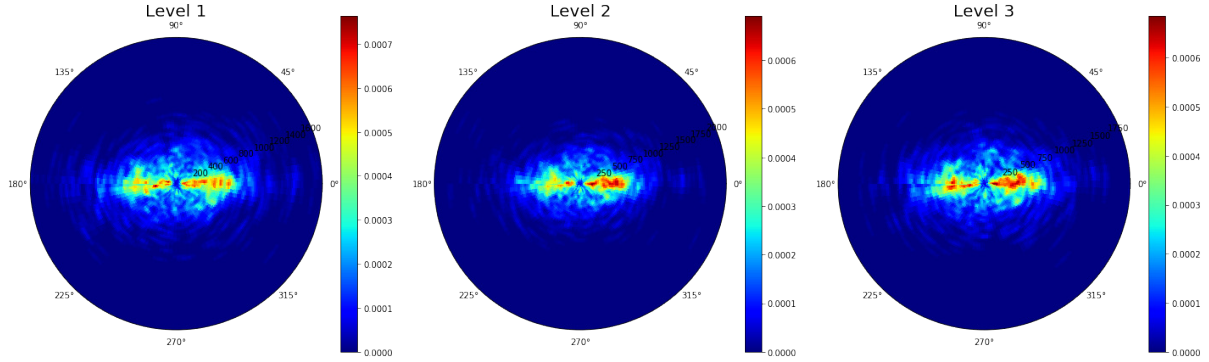


Figure 22: Joint saccade on different distortion types

Not surprisingly, the polar plot above has also shown that there is a strong bias movement on 0° and 180° with short distance action.

5.4 Result Summarisation

The data across different comparison group have shown a real similarity between their rivals. Overall from the histogram analysis, all participant's saccade amplitude and orientation result have a very similar data shape. The overall data have shown a trend in saccade amplitude analysis where the participants were preferred to have a short distance movement while different distortion types and levels will not have any significantly affects on how their eye behaviour were. The same applies to their saccade orientation result, where the overall data shape was the same. Participants preferred the peak through certain angles, -180 , 0 , 180 degree. Once again, distorted images will not affect how their viewing tendency were.

Such difference on the result might vary on how human and machine visual an image and recognise objects. Machine treat image as pixels, they do not have the concept of object,

everything in computer's eyes is just numbers or signal, where human recognise object as it overall shape and form. A potential explanation on such result is because machine was really sensitive with numbers and variable, slight changes may vary the result in huge manner. Whereas human recognised object as an overall shape of it, those distorted effect used in the project does not really changes the general shape on the object, where this may not affect how human may approach to the image.

References

- Alemdag, E. & Cagiltay, K. (2018), ‘A systematic review of eye tracking research on multimedia learning’, *Computers & Education* **125**, 413–428.
- Andrew Go Lee, Ashwini Kini, B. A. O. (2020), ‘Saccade - eyewiki’.
- Blignaut, P. (2014), ‘Mapping the pupil-glint vector to gaze coordinates in a simple video-based eye tracker’.
- Britannica, E. (2015), ‘Distance formula’. Accessed 10-05-2020.
URL: <https://www.britannica.com/science/distance-formula>
- Cassin, B., Rubin, M. L. & Solomon, S. (1984), *Dictionary of eye terminology*, Vol. 10, Triad Publishing Company Gainesville.
- Djamasbi, S., Siegel, M. & Tullis, T. (2010), ‘Generation y, web design, and eye tracking’, *International journal of human-computer studies* **68**(5), 307–323.
- Few, S. (2013), ‘Data visualization for human perception’, *The Encyclopedia of Human-Computer Interaction*, 2nd Ed. .
- Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N. & Wichansky, A. M. (2002), Eye tracking in web search tasks: design implications, in ‘Proceedings of the 2002 symposium on Eye tracking research & applications’, pp. 51–58.
- Grillon, H., Riquier, F., Herbelin, B. & Thalmann, D. (2006), Use of virtual reality as therapeutic tool for behavioural exposure in the ambit of social, in ‘International Conference Series on Disability, Virtual Reality and Associated Technologies (ICDVRAT)’, number CONF.
- Hessels, R. S. & Hooze, I. T. (2019), ‘Eye tracking in developmental cognitive neuroscience—the good, the bad and the ugly’, *Developmental cognitive neuroscience* **40**, 100710.
- Hooze, I., Holmqvist, K. & Nyström, M. (2016), ‘The pupil is faster than the corneal reflection (cr): Are video based pupil-cr eye trackers suitable for studying detailed dynamics of eye movements?’, *Vision research* **128**, 6–18.
- Hoppe, S., Loetscher, T., Morey, S. A. & Bulling, A. (2018), ‘Eye movements during everyday behavior predict personality traits’, *Frontiers in human neuroscience* **12**, 105.
- Jacob, R. J. (1990), What you look at is what you get: eye movement-based interaction techniques, in ‘Proceedings of the SIGCHI conference on Human factors in computing systems’, pp. 11–18.
- Kapoula, Z., Groner, R., Menz, C., Fisher, D. & Monty, R. (1983), ‘The influence of peripheral preprocessing on oculomotor programming’, *Eye movements and physiological functions: International views* pp. 101–114.
- Koehler, K., Guo, F., Zhang, S. & Eckstein, M. P. (2014), ‘What do saliency models predict?’, *Journal of vision* **14**(3), 14–14.
- Le Meur, O. & Coutrot, A. (2016), ‘Introducing context-dependent and spatially-variant viewing biases in saccadic models’, *Vision research* **121**, 72–84.

- Le Meur, O., Coutrot, A., Liu, Z., Rämä, P., Le Roch, A. & Helo, A. (2017), ‘Visual attention saccadic models learn to emulate gaze patterns from childhood to adulthood’, *IEEE Transactions on Image Processing* **26**(10), 4777–4789.
- Le Meur, O. & Liu, Z. (2015), ‘Saccadic model of eye movements for free-viewing condition’, *Vision research* **116**, 152–164.
- Mahdi, A. & Qin, J. (2017), ‘Deepfeat: A bottom up and top down saliency model based on deep features of convolutional neural nets’, *arXiv preprint arXiv:1709.02495*.
- Mathôt, S. (2018), ‘Pupillometry: Psychology, physiology, and function’, *Journal of Cognition* **1**(1).
- Merchant, J. (1967), ‘The oculometer’.
- Nattkemper, D. & Prinz, W. (1987), Saccade amplitude determines fixation duration: Evidence from continuous search, in ‘Eye Movements from Physiology to Cognition’, Elsevier, pp. 285–292.
- Obaidellah, U., Al Haek, M. & Cheng, P. C.-H. (2018), ‘A survey on the usage of eye-tracking in computer programming’, *ACM Computing Surveys (CSUR)* **51**(1), 1–58.
- Purves, D., Augustine, G. J., Fitzpatrick, D., Katz, L. C., LaMantia, A.-S., McNamara, J. O., Williams, S. M. et al. (2001), ‘Types of eye movements and their functions’, *Neuroscience* pp. 361–390.
- Rai, N. (2019), ‘Analysis of machine learning models for visual saliency’.
- Research, S. (n.d.), ‘About eye tracking’. Accessed 26-04-2020.
URL: <https://www.sr-research.com/about-eye-tracking/>
- Saabas, A. (2016), ‘Histogram intersection for change detection’. Accessed 28-04-2020.
URL: <http://blog.datadive.net/histogram-intersection-for-change-detection/>
- Termsarasab, P., Thammongkolchai, T., Rucker, J. C. & Frucht, S. J. (2015), ‘The diagnostic value of saccades in movement disorder patients: a practical guide and review’, *Journal of clinical movement disorders* **2**(1), 14.
- Tobiiproh (n.d.), ‘What is eye tracking?’. Accessed 26-04-2020.
URL: <https://www.tobiipro.com/blog/what-is-eye-tracking/>
- Was, C., Sansosti, F. & Morris, B. (2016), *Eye-tracking technology applications in educational research*, IGI Global.
- Zhu, G., Wang, Q. & Yuan, Y. (2014), ‘Tag-saliency: Combining bottom-up and top-down information for saliency detection’, *Computer Vision and Image Understanding* **118**, 40–49.