Dynamic Classification of Eye-Tracking Movements: Usage and Ranking

Josef Jan Krivan

Thesis submitted for the degree of
Master in Applied Computer and Information Technology (ACIT)
30 credits

Department of Computer Science
Faculty of Technology, Art and Design

OSLO METROPOLITAN UNIVERSITY

Spring 2022

Dynamic Classification of Eye-Tracking Movements: Usage and Ranking

Josef Jan Krivan

Abstract

Eye-tracking including its uses, research and progression has seen much growth over the years, especially in the medical field. With this naturally comes many challenges, such as proper application in numerous fields as well as the directions to take the technology towards and what is best to focus on with such a versatile field.

In our case, we discuss the randomness of eye-patterns, the distinguishing of the different eye-movement types and how best to categorize these movements correctly. This research is especially important since, the correct classification of these distinct eye movements is vital for fields such as medicine and psychology, specifically in the field of medical diagnosis.

The main methods for distinguishing these eye features is through the use of algorithms, designed to find patterns in movement properties, eventually classifying the features. The research for this thesis as well as its writing was done entirely between January and May 2022. In this time, different approaches were used, first by analysing the eye-tracker data with its corresponding algorithm, and then moving onto robust, established methods within the field such as velocity and dispersion threshold-based algorithms.

Through testing, it was determined that the basic I-VT velocity based algorithm turned out to be the best result, despite being the most simplistic algorithm implemented.

Acknowledgments

I would like to thank * Pedro Rego Lencastre & Pedro Lind * for their contributions towards this thesis regarding the structure, writing and implementation of the work.

Contents

Chapter 1

Introduction

Eye-tracking is the recording and studying of eye movement based on specific visual stimuli, either in the form of static or moving objects of interest. It is an important part of ophthalmology, studying the human eye in regards to how it acts and moves. While the study of visual perception, attention, cognition and related eye movements originated in the 1960s, advances in technology and methodology during the 21st century have propelled the eye-tracking industry forwards, opening up a wide range of application areas and uses [1]. There are many ways of studying our eyes, but what will be focused on in this thesis is the monitoring of movement and resultant behaviour based on visual stimuli.

Many different domains, especially in research studying behavior, such as areas in psychology or human development, utilize eye-tracking. Some of these areas include image scanning, driving and reading [2]. Arguably the biggest application domain is within biomedical research and clinical diagnostics. Eye movements can be used to diagnose Alzheimer's, HIV-1, schizophrenia and attention deficit [3, 4]. Other, typical applications of eye movement research are notable as well, such as its use in education. Such examples include a 2014 computer education project regarding its use "as an instrument for computer science education research "[5], as well as a 2005 study about how students attend to photographs within science related PowerPoint presentations [6]. Another growing domain regarding eye-tracking is in the realm of visual marketing regarding commercial interests. Such studies include a 2006 research project focused on consumer behavior regarding how point of purchase is influenced memory and attention-based visual factors [7]. Additionally, a 2008 paper tackles eye-movement and visual attention to discuss emerging issues within visual marketing [8].

All of these eye-tracking studies and experiments, at least more recently, are done with sophisticated technology. This involves actual eye-tracking hardware where eye-movements are recorded digitally at varying frequencies, and corresponding software to view the results. A prominent example are Tobii's eye-trackers, demonstrating different models of eye-trackers as well as sophisticated GUIs (Graphical User Interface) to view results with [9]. One of the main goals of eye-tracking is to distinguish the different features/events during eye movements. The two main categories of eye movements are *fixations*, points of interest where the eye focuses on in an image, and *saccades*, the eye movements between fixation where the eye is only travelling and not focused on anything specific. These are not necessarily features, but are rather classifications based on features. Other features also include *smooth pursuits*, a different kind of fixation where the object of interest is in motion. *blinks*, where the eyes are not being recorded, thus resulting in data loss. *noise*, the unfiltered data that is a result of high frequency and a hindrance to accuracy and *post-saccadic oscillations*, defined originally as "glissades"[10] are eye-movements prior at the end of a saccade period where the eye wobbles during deceleration onto a fixation point. The main problem being faced in this research is to determine how one can distinguish between these events, mainly fixations and saccades. This is done through the use of mathematical algorithms that approach the data in different ways and conditions to aid in the separation of these features. Due to the numerous approaches and algorithms that can be utilized, the question that will be investigated is "Which algorithms are best suited for the classification of eye-tracking features and movements?". Some algorithms use the velocity of eye-trajectory (I-VT), while others might use dispersion (I-DT) based on clusters of gaze-points. More advanced approaches include the use of Hidden Markov Models (I-HMM), Minimum Spanning Trees (I-MST), duration-based area of interest (I-AOI) algorithms [2], algorithms based off Lyapunov experiments [3] as well as algorithms utilizing machine-learning [11]. These different algorithms will be evaluated based on implementation difficulty, complexity, usability/versatility and classification effectiveness with a dataset created specifically for this research.

The purpose of this study is to find out what algorithms and data-analysis approach is best for the eye-tracker data that will be collected. The clear observation that will be made later is that certain algorithms will work better than others for this specific data, while in other eye-tracking experiments different algorithms will be more suited for that task. As stated, this study is mainly for the purpose of researching the many different mathematical algorithms and approaches, evaluating their technical performance, implementation difficulty, and if possible, how applicable they are for specific tasks, thus discovering why some work better than others. A hypothesis can be made that the more advanced algorithms, while more complex and difficult to implement, and perhaps not as flexible or usable as more general approaches, will end up producing not only superior, but more classification results with more features.

As mentioned previously, there is good reason to be conducting such experiments when it comes to filtering features in eye-tracking. Technical advancements combined with a comprehensive understanding of the technical data from eye trackers will greatly improve the diagnosis within medical research and human psychology. Implementation of effective feature separation algorithms will further research by providing more effective telemetry, resulting in more readable data. Several different assumptions can be made; Many of the algorithms, especially those of less complexity, may end up only being proficient at detecting one kind of feature (e.g. saccade or fixation distinction). The more complex algorithms will be able to detect the auxiliary features (e.g. blinks or post-saccadic oscillations). Noise will inevitably present a problem with feature filtering, therefore a measure of noise reduction is necessary. Another cause for data loss are blinks, which will also negatively affect the results. There exist methods for filling in the blanks, but these will never be completely accurate. Lastly, some algorithms may simply not work well with the kind of eye-tracking data that will be utilized. Therefore, seemingly more complex algorithms that may produce better results are perhaps too specialized to be of use with this data.

The main importance of this problem is the distinguishing between fixations and saccades. We focus on these two features as the other mentioned features (e.g. blinks and noise) are expected background features that result in skewed or lost data. Smooth pursuits (moving fixations points) are generally studied in separate experiments from static fixations, however some studies have tackled the issues of detecting smooth pursuits within fixations and saccades [12].

This thesis will be divided up into 4 chapters. In the first chapter, the state of the art, including notable previous work and more modern novel approaches, as well as other fundamental concepts such as the human eye, statistical models and eye-tracking data will all be discussed. In the second chapter, the data analysis from the original experiment, as well as the algorithmic implementations will be discussed. In the third chapter, the results will be shown, evaluated, and dicussed. In the fourth and final chapter, a conclusion will be made, with evaluation against original problem, and future work will be suggested.

Chapter 2

Background and State of the Art

2.1 The Eye

The human eye is a complex organ in the human body, consisting of several layers of components (e.g. pupil, cornea, iris etc..) and muscles to operate. The basic idea of our eyes is identifying objects and identify visually distinct entities within one's own vision. We rely on light reflecting off objects and thus travelling into our eyesight. The visual signals entering our eyes are then sent to the brain for processing, reaction and interpretation [13]. We see light with two types of visual receptors in the retina, rods and cones. Rods are used for lower light areas and create black and white images from objects while cones are for brighter areas and produce colour from objects. There are many more rods than cones in our eyes as there are approximately 20 rods for every cone [14]. Soussan Djamasbi describes the fovea, a small point in the middle of the retina that produce a much sharper image than the rest of our eyesight. Most of the cones in our eyes are located in the fovea which is why we must look directly at objects that are to be focused on. As it only covers about 2 degrees of our field of vision, we must constantly adjust to focus on several different objects in the field of view, hence the high frequency of saccades and fixations during regular eye operation [13].

In figure 2.1 included on page 6, one can see the basic structure of the human eye. This includes components such as the lens, retina, iris and fovea from the previous definitions, stating where they are located within the eye.

Studying the eye to any degree is important, as eye-movements are some of the most frequent movements regarding the human body. This is simply due to the fact that visual information is everywhere and our eyes are attracted to objects, switching constantly from fixation to fixation [15]. Due to the importance of the eye, explaining the movements, behaviors

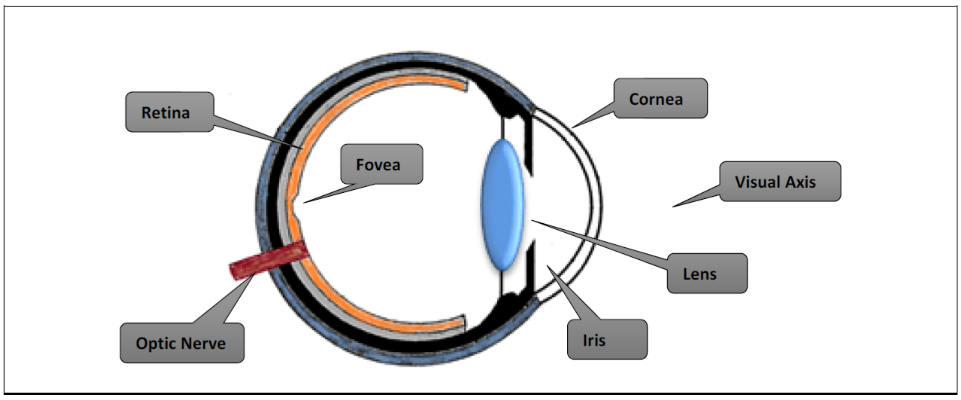


Figure 2.1: An illustration of the human eye [13].

and patterns of our eyes cannot simply be undertaken by mathematics or medicine alone. Mathematics allows the analysis of the eye from a technical level, identifying said patterns and movements, but it cannot explain why these movements occur, which is where the medical studies of the eye excel. Certain discoveries, such as post-saccadic oscillations, caused by over or undershooting fixations, were first observed mathematically by Nyström & Holmqvist via their filtration algorithm [10]. Eye behavior such as wobbling effects like this are most likely easier to explain from a medical perspective, as the eye is a compressible object and imperfect movements are apparent. Interpretation of the technical data aids in the medical applications of eye studies and tracking. As stated, eye-tracking has many application areas including Human-Computer Interaction (HCI), medical diagnostics, psychological studies and computer vision [16].

2.2 Statistical Learning Models of Time Series

Data modelling for eye-tracker data tends to follow many standardized ways for the processing and analysis of results. Many sources start with raw data from eye-tracking experiments, collected through the use of stochastic process, gathering the positional gaze data, and other movement data. Afterwards, pre-processing will be conducted (e.g. exclusion of unnecessary data, such as columns in a data set, removing redundancy or splitting up data sets to improve management). This process may include some additional tasks such as removing noise, filling in lost data points (e.g. from blinks) or visualizing the data. Then filtering algorithms, along with any machine learning algorithms, are applied to the cleaned data as models to be evaluated. This is usually carried out through the use of statistical measurements, such as accuracy or precision results, especially since eye-tracking filtering is usually a classification task. Additionally, the model results are displayed on a table and can also be visualized as a means of getting a grasp for the results of any specific classification test.

There are many different ways to model data and present it. Some methods are text-based methods, using tables and values to display results. Since the eye-tracking problem in this thesis is related to separating features, it is a classification task, therefore the textual results will be related to this specific type of problem. Tables are likely to include accuracy, precision, recall or f1-score along with the raw classified data values from the total amount of data points in the set for each algorithm used. Another type of table used often is a matrix. This is either used to directly compare models or columns with one another, or used in confusion matrices. The confusion matrix is highly relevant as it displays the raw classification results, showing the amount of results that are true/false positives and true/false negatives. It is relevant to any classification problem, as any classification results (e.g. precision or recall) and the formulas for calculating the results come directly from the confusion matrix results.

Besides text based tables, there also exist a few ways of visually modelling results using graphs or maps. One of the most common types of visualizations are two dimensional time series graphs, due to the nature of eye-tracking and recording eye movements over a certain period of time. Typically, a time measurement (e.g. milliseconds, frequency (Hz) of the eye-tacker) or duration runs along the x-axis, while some kind of positional or movement data (e.g. X/Y positions, X/Y changes, velocity) runs across the y-axis. Another typical visualization are two dimensional scatter plots, that usually reflect the actual data that was recorded. Generally, the gaze positions along the X-axis and Y-axis are the two visualized values, showing the actual position of the eyes across the visual space of the experiment. These kinds of visualizations are usually enhanced with colour, segmenting fixations and saccades, or even turned into heat maps, showing the intensity of which parts of the visual space were being gazed at. There also exist other unusual visualizations, such as histograms for seeing how certain data points occur inside of a full series.

There are many examples of the models, tables and graphs described in this section for the purposes of analysis and results in section 2.4.

2.3 Eye-tracker Data

The first step in the data processing is the retrieval and collection of eye-tracker data as is the most vital component of this study and is necessary to conduct any algorithmic tests for the purposes of usage and evaluation. From a broad perspective, there exists two distinct approaches to data collection, offline retrieval and online retrieval. Offline data collection refers to eye-tracking experiments that were done on-site with physical equipment inside a physical environment. This data is to be collected from a physical eye-tracking device (e.g. a Tobii eye-tracker) and analysed post-experiment. Online retrieval refers to experiments done from a remote location, where the tests are done with software as opposed to hardware. These tests are also done in real-time. The data transmission happens during the test, where algorithms and filters are being applied

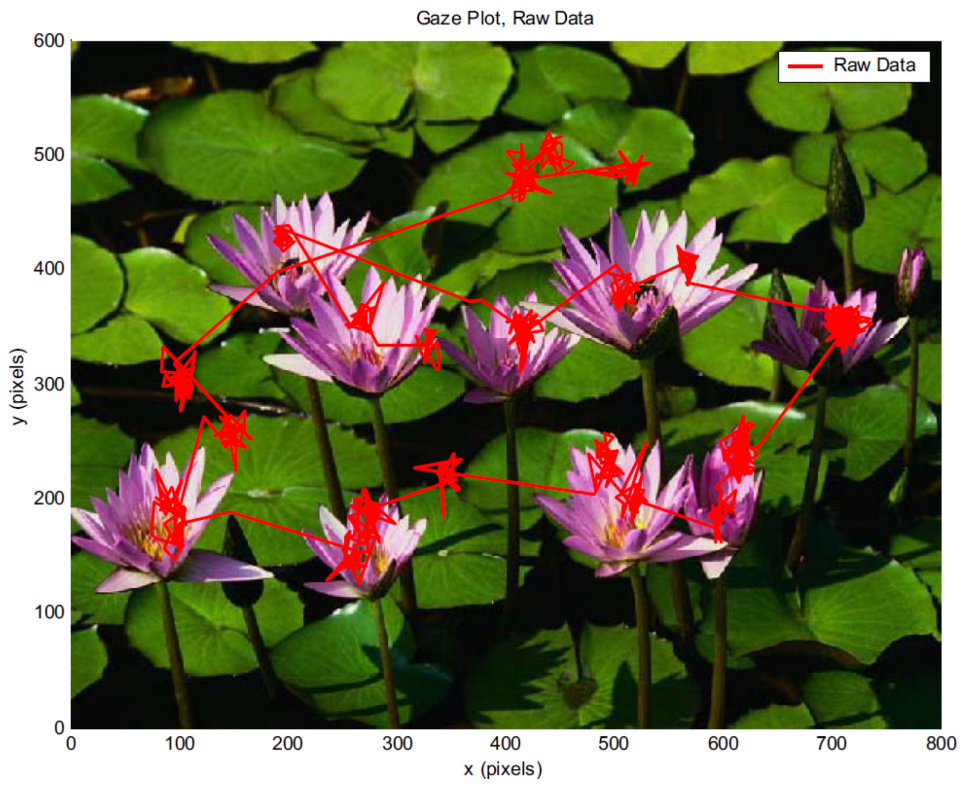


Figure 2.2: Raw gaze data with the background image used in the experiment [14].

simultaneously. The clear downsides however, is the higher rate of data loss due to poor latency/packet loss, higher overall latency and lower frequencies of the data capture due to software limitations.

Many different vital components for processing are collected in the raw data files from the collection phase. First are the timestamps, frequency based (ticks) and time based (seconds) as well at total duration statistics. Second are the positional statistics, such as gaze positions along both axis of the 2D space, additionally including the dimensions of the display (e.g. 1920x1080). Additionally, the gaze trajectories and directional changes for each individual eye (as well as combined) are recorded. Third is the hardware and software information, mainly for the purposes of context or comparison. These may include the name of the sensor being used as well as its accompanying software, date of the experiment, what filters (e.g. a fixation algorithm) are being used, and participant ID.

The second major task for the data processing is the actual classification of the features to be obtained. There are again, two prominent approaches manual and automatic feature classification.

The first of what can be simply identified as human classification. This is usually done by experienced human coders, who can either be trained or untrained to read eye tracking data. One issue with this approach are the proficiency levels of those who will be classifying the data, and human improvement mostly comes from experience. Another major issue is the unreliability of humans in-general. This is due to the fact that humans may interpret data in different ways. One coder may think certain patterns are saccades, while another may classify the data differently simply due to a different approach in observation or interpretation, which is more of a physiological issue rather than a technical one. One can always observe the general behaviors of classification of eye-tracking data with experienced, trained coders against those who are inexperienced or untrained.

The second, more straightforward approach is automatic classification, done by machines and software. Pre-written algorithms and software attempt to automatically classify data that is fed to them. This is based on some kind of conditions related to the raw data from the eye trackers, whether it be positional, time-based data or even data calculated after the experiments such as velocity. This is the general approach that will be evaluated in this study, and the different methods and algorithms will be explored and demonstrated in greater detail later on.

In figure 2.2 included on page 8, one can see an eye tracking experiment that was previously carried out. This contains the background image that the participant was observing for analysis, as well as the raw gaze plot in the foreground, marked with a red line. Looking at this particular experiment, it is clear where the fixations and saccades are on the gaze plot, where the travel times (straight lines) are saccades and blobs of small red lines are fixation points.

2.4 Previous Works in Eye-Tracker Data Modelling

Here, we will discuss the existing applications and works of the algorithms described earlier, either functioning in tandem with each other or developments built off of the core fundamentals of those algorithms. In addition, some studies also focus on the auxiliary eye-tracking data, either by focusing on the alternative classifications (e.g. blinks or noise), combining said classifications that would otherwise be in separate data streams, or by creating new features that branch from the existing ones.

Interestingly, some studies go against the idea of using algorithms and automated systems to distinguish features and filter results. Some believe that manual detection via human observation is still a prime option.

In 2018, Hooge conducted a study to evaluate if "human classification by experienced untrained observers" was the best approach for detecting fixations in eye gaze data. Their research showed classification differences in fixation duration and number, therefore concluding that it is not the best approach [17]. While not completely decisive, perhaps a hybrid approach combining machine algorithms and human detection (human-in-the-loop) may end up being the most effective solution, but perhaps not the fastest approach. The algorithms' goals is to make the separation process more accurate, faster and efficient, therefore a focus on accuracy with the goal of eliminating human input could be emphasized.

Many different studies use variants of the I-VT and I-DT based algorithms as they are considered easier to implement as well as fundamentally robust and can be built upon. Salvucci and Goldberg did a comprehensive evaluation of these algorithms, going into detail on how the algorithms work on a fundamental level, as well as outlining them in a technical level via a pseudo code process. They also evaluate the algorithms based on their accuracy, speed, robustness, implementation difficulty and parameter count [2]. In addition, they implemented dispersion (I-DT) and area of interest (I-AOI) algorithms into their own interactive environment for eye-tracking protocols known as EyeTracer [18]. Andersson et.al also does a review of algorithms. He describes 6 additional algorithms, CDT (Covariance Dispersion Threshold), EM(Engbert & Mergenthaler velocity threshold), IKF(Identification by Kalman Filter), NH(Nyström & Holmqvist algorithm), BIT(Binocular-Individual Threshold) and LNS(Larsson, Nyström & Stridh). From their results, it appears that apart from NH and LNS, the existing algorithms are only made for detecting fixations and saccades and do not detect post-saccadic oscillations, smooth pursuits or blinks. These systems must be implemented either manually or with separate algorithms. Manual human operation however, can detect these events, but will likely not be as accurate as a machine approach. No real conclusion is made, as the algorithms vary in results [19].

Introduced earlier, there exist two distinct methods of capturing eye-tracker data. One can either capture data offline or online. Offline data capture refers to experiments where the subjects are physically in the same place as the experiment, using eye-tracking hardware to conduct tests. Online data capturing (i.e in real-time) is when the participants complete the experiments remotely over the internet. A few studies have addressed the issues with online versus offline detection, as well as filters for both approaches. In 2019, Schweitzer and Rolfs present a "velocity-based algorithm for online saccade detection", combating the issues of late detections and false alarms. Their adaptive algorithm was able to achieve high detection accuracy with a false alarm rate of under 1%, combined with latency as low as 3 ms [20]. Additionally, Olsson also tackled the issues of real-time and offline filters in 2007, comparing filtering techniques for eye-movement (e.g. controlling a mouse cursor) / gaze-data in real-time and post-processing (offline) scenarios [14].

As a technical showcase, Anneli Olsen combines a filter and software approach, introducing a a more advanced I-VT filter that fills in-gaps where data is lost (e.g. blinks), selects a distinct eye, applies noise reduction and calculates the velocity for the eventual I-VT classification. In addition, a novel GUI approach is utilized via Tobii studio along with various Tobii eye-tracker models with varying frequencies [9].

There also exist other studies to introduce relatively impressive technical algorithms or inclusions to the filtration process. In 2013, Tavakoli et.al incorporated saliency estimation into a saccadic eye-movement simulation model. They apply a stochastic filtering framework for said saccade generation. The model then dynamically produces saliency maps based predicted fixations from the generated saccades, which performs favourably to other saliency models [1]. Krejtz et.al demonstrates eve movement characteristics and their variability. He introduces a novel technique by quantifying the transitions between areas of interest using gaze switching patterns, modeled as markov chains. Additionally, he quantified the complexity of the switching patterns (Shannon's entropy coefficient of the Markov Model) to determine the attention distribution over several AOIs [21]. In 2009, Pieter Blignaut conducted a threshold test by evaluating the sensitivity of the I-DT algorithm (an enchanced algorithm from to Salvucci and Goldberg)[2] where the difference of scan path between the points of regard (POR) are used to determine optimum threshold values [22]. Korda et.al presented a rather advanced technique of automatic eye movement identification in 2017, using a technique known as the Largest Lyapunov Experiment (LLE) and the logarithm of divergence. This technique can be used for the automatic identification of saccades and blinks. The relatively high performance of their model compared with 2 other filtering methods (I-VT and the Petterson et.al blink-identification method from 2013) is benefited by the fact that LLE is used to identify and predict various diseases, therefore bringing more meaningful use to the field of diagnostics within eye-tracking [3].

Mentioned previously, some studies have attempted to filter saccades rather than focusing on fixations. Larsson et.al presents a novel approach, comparing manual methods with a velocity based algorithm, with regard to acceleration, when detecting saccades and postsaccadic oscillations in smooth pursuit experiments [4]. As stated earlier, some studies combine smooth pursuits and static fixations. Komogortsev amd Karpov in 2012, attempted to classify and score smooth pursuits within "the presence of fixations and saccades", which they call "Ternary eye movement classification". They utilized an algorithm combining velocity and dispersion methods resulted in effective classification results [12]. A well known 2010 research paper by Nyström and Holmqvist were perhaps among the first to identify these oscillations, defined as *glissades* in their research. These movements are a period where the eye wobbles prior to fixating on a visual object. A new velocity-based approach was able to identify these movements as they occur about half of the time during saccades, showcasing these features as non-trivial [10].

When it comes specifically to data modelling and the whole data analysis process, many papers showcase different ways of processing, analysing, implementing and displaying the results of their tests.

As mentioned previously, tables are a good way of displaying relevant data, statistics and results in a compact, readable way. Many papers use a combinations of tables and matrices. A straightforward example of a easy-to-understand table is the "summary of identification methods" presented in Salvucci and Goldberg's 2000 summary of existing fixation identification algorithms. It clearly demonstrates the ability of each algorithm through the set criteria of accuracy, speed, robustness, ease of implementation and the amount of parameters required, through the use of check marks for positive results, and crosses for negative results [2]. Hooge et.al, studying human classification in fixation detection in 2017, has an interesting use of tables to convey important information. Table 1 shows details of "the 12 experts who classified fixations" detailing their names, age, years of expertise, the subject group they were classifying from, the algorithms they utilized, and what eye-tracker was used. Such information is obviously crucial when the paper is focused around experienced observers in the field of fixation classification [17]. Confusion matrices are seen less often, but still provide vital data for result interpretation. In 2016, Andersson et.al makes good use of them in his evaluation of "eye movement event-detection algorithms". He displays confusion matrices for different kinds of evaluation such as for static images, moving dots, and videos. These contain all of the evaluated algorithms (IDT, IVT, IHMM, IMST and others), comparing them with ratio (the rate of which the specific algorithm disagreed with human evaluation), and the maximum/maximum error for fixations, saccades, smooth pursuits, blinks and more. Such thorough information makes evaluating the algorithms more intuitive and less confusing [19].

The other, more engaging way of showing analysis and results is through different types of graphs and maps. They usually come in the form of scatter plots or time-series graphs. Maps generally include heat maps or raw eye-gaze positions on a relevant background. A typical visualization of data is comparing eye-gaze positioning for x and y data, or positioning relative to time. Schweitzer and Rolfs' 2019 paper on an adaptive algorithm demonstrates different saccade detection techniques, showing X-axis and Y-axis gaze positions with different approaches such as the spatial-boundary technique, absolute-velocity threshold, as well as their own proposed algorithm for saccade detection [20]. Anneli Olsen in his 2012 study on "the Tobii I-VT fixation filter" demonstrates a comparison between x-axis gaze data and velocity, directly comparing the parallels between eye movements along an axis, with the velocity at which it was measured. Later on, he also compares x-axis and y-axis gaze points, using distinct shapes for fixations and saccades [9]. In 2000, Dario D. Salvucci also visualized a sample of gaze data using velocity-based and fixation highlighting, "numbered fixations with target associations" and colour-based fixations for predicted and predicted fixations [18]. In 2012, Komogortsev and Karpov demonstrate intuitive time-series graphs in their paper regarding the automatic classification of smooth persuits, while fixations and saccades are present in in any one eye-tracking experiment. The graphs intuitively shows I-VDT (a velocity-based smooth pursuit filter) classification results, dintinctly marking out saccades, fixations, smooth pursuits and noise for eye movement on the x-axis along a time-series

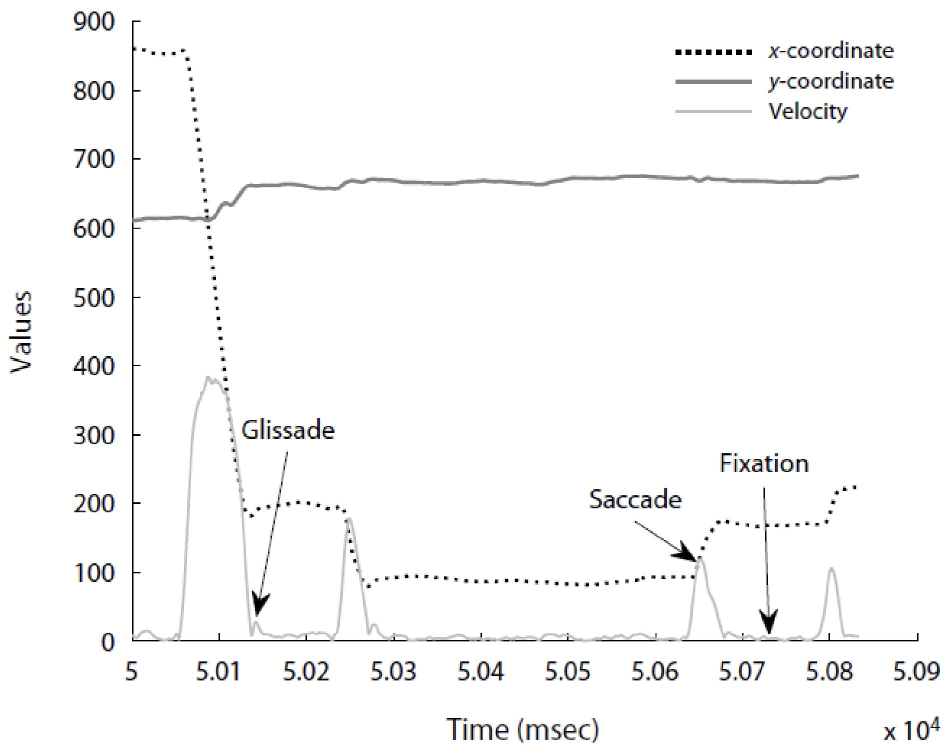


Figure 2.3: X/Y gaze coordinates over a period of time, labelled [10].

graph. These graphs are used to compare the I-VDT algorithm to manual classification [12]. Nyström and Holmqvist also use time-series graphs in their 2012 paper. In a simple example, they compare x-axis and y-axis eye-movements and velocity on a single time series graph for the purposes of demonstrating how "glissades" (post-saccadic oscillations) work. They point out in the time-series where each of these events occur by observing the velocity value over a fixed time [10].

In figure 2.3 included on page 13, one can see the graph containing x-axis and y-axis eye gaze values, as well as the overall velocity of the eye movements over the course of a defined amount of time. This is of a time-series graph used by Nyström and Holmqvist described above, where the details of eye-movements and their corresponding velocity are shown in the legend above. Additionally, they included indicative text to mark fixations (low velocity ranges), saccades (higher velocity ranges) and glissades (small eye movements made at the start of most fixations).

The last notable ways is visualizing results are through histograms and heat maps. In 2021, Lencastre et.al presented a database for eye-movement recordings including some relevant visualizations of the raw data. Histograms are used to show how the fixations and saccades are occur in the given series, comparing the occurrences of fixations and saccades in a single series by separating them into individual histograms. Heat maps are also used, displaying eye-gaze trajectories for the purposes

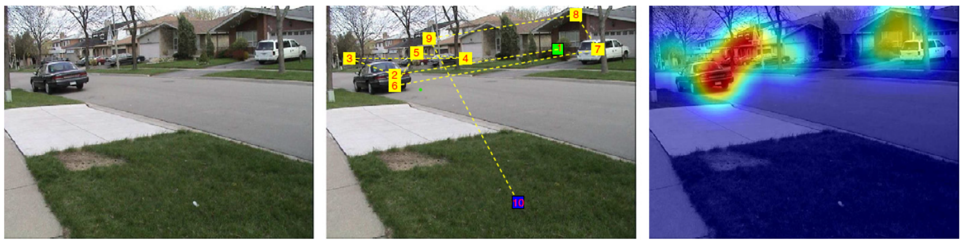


Figure 2.4: Generated heat maps from predicted gaze data [1].

of assessing "visual communication using a letter board" [23]. Tavakoli et.al in their 2012 study on a saliency-based approach for fixation predictions and saccade generation use a variety of eye-gaze trajectory maps and heat maps. They showcase a saliency map with generated fixations, first showing the raw image to be observed in the experiment, then showing the predicted fixations and saccades with the gaze trajectories generated and lastly showing the resulting saliency map using the predicted fixations from the earlier generated gaze data. They also demonstrate this "generated saccade sequence" on several more real images, comparing their model with 4 different human saccade sequences [1].

In figure 2.4 included on page 14, one can see the process for which the saliency heat maps were generated. First, there is the reference image (similarly seen in figure 2.2) that their prediction system will utilize. Second, fixations predictions are generated on the image based on expected points of interest. Lastly, the saliency mapping from the generated fixation points creates an interesting heat maps, where the predicted points of fixation will be, and to an extent, which fixation points are most relevant in the specified image.

2.5 Ethics & Methodology

Last year, Samip Bhurtel et al. published a paper regarding the promotion of "empathy towards non-verbal people", via the usage of eye-tracking. With their use of eye-tracking, they collected raw data from their experiments which will also be used in this thesis [24]. This data included in both the conference proceedings paper, as well as the full thesis, is approved for use by the NSD (Norsk senter for forskningsdata) as it complies with Norwegian research and ethics standards. Therefore, this thesis complies with Norwegian ethics and research board, as the data being used for the experiments has been approved prior to the writing of this thesis.

This being stated, the mentioned data from last year will be used for all implementation in this thesis, however, if the new proposed data for an upcoming eye-tracking test is to be approved for use, the aforementioned data from above is no longer valid. This will most likely be the case since the density of the current data is too low as there are too few data points for accurate testing. As it stands, the methodology involves using the current data for the purposes of data analysis, algorithmic implementation, testing and interpreting results with tables and graphs.

2.6 Algorithms and Approaches to Distinguish Between Fixations and Saccades

In this chapter, the algorithms mentioned earlier, some of which were used by previous authors will be introduced properly. The fundamentals will be discussed, how they generally work and what parameters are involved. Additionally, the implementation of them will be discussed and a process will be demonstrated later.

There are several methods in which one can classify features of eye-movement. Many of the significant ones already mentioned will be discussed in this chapter, along with any potential novel approaches that combine approaches or add unusual variables. In this chapter, the different general approaches will be introduced, along with any corresponding or acknowledged algorithms using said approaches. A through analysis will be conduced on these approaches and algorithms and a general verdict will be made prior to implementing them at a later stage.

2.6.1 Velocity-Based Algorithms

One of the most fundamental, intuitive and common approaches for fixation detection is filtering the data using a velocity-based algorithm. Velocity based algorithms are as they sound, taking advantage of eye-tracking velocity to classify the different features over the course of a session. This is mainly done by characterising high velocity movement as saccades and relatively low velocity movement as fixations. This method ideally assumes that the time between each eye-gaze point is constant, therefore velocity is determined exclusively by the distance between points. The main challenges with this method is that the sampling rate of real experiments may not necessarily be constant and thus give inaccurate velocity calculations. This approach can be deemed most simple as it does not rely on any data (e.g. relative duration) other than relative gaze-points for classification[2]. As this is a well known approach, many different algorithms exist, basing themselves on the general technical methods that will be discussed next.

Identification via Velocity Threshold

The first velocity-based algorithm is Identification via Velocity Threshold or I-VT. The core concepts that apply to velocity-based algorithms on the whole apply entirely to I-VT as it is the most basic and fundamental velocity-based algorithm currently existing. This makes the I-VT algorithm easy to understand. Due to its simplicity, it is generally used as a basis for more advanced algorithms such as the Hidden Markov Model algorithm that will be introduced later. Other Algorithms such as Nyström and Holmqvist's model also builds upon I-VT as a basis for glissade detection.

The I-VT algorithm revolves around separating and classifying fixations and saccade samples using a fixed velocity threshold. If the calculated velocity between two points is below the threshold, it will be classified as a fixation, while sequences above the threshold will be classified as saccades [19]. Due to its simplicity, it results in a "straightforward and robust" algorithmic approach. The velocity threshold in I-VT generally remains static, thus adaptive thresholds remain unnecessary due to "strong physical and physiological underpinnings" in velocity profiles [2].

As inferred from the algorithmic properties defined above, I-VT is the simplest identification algorithm to both understand as well as implement. The velocity measurements for point-to-point eye-tracking are generally defined using degrees per second $(^{\circ}/_{s})$, also known as angle-velocity. For example, a low velocity of $(^{50^{\circ}}/_{s})$ would be a part of a fixation, while a high velocity of $(^{350^{\circ}}/_{s})$ would be apart of a saccade.

The steps required to implement the algorithm are relatively simple and straightforward. First, one must calculate the velocities (angle-based or otherwise) between each point for the whole procedure, on a point-to-point basis. The velocity is calculated based on the current point in the increment (p) from the previous point (p-1) in the increment. Using a vector-based approach, this would mean calculating the vector based on the combined x-axis and y-axis movements. Second, using a predefined velocity threshold, the algorithm will determine if the series of points are a part of a fixation or a saccade (below threshold for fixations, above threshold for saccades). Once a series of fixations between consecutive points are made, they will be clumped into a fixations group, and all points considered as saccades between said fixations will be excluded. These fixation groups are then translated "to a representation <x,y,t,d> as the centroid" (center of mass) by the algorithm. The points x and y indicate the centroid of a fixation group (e.g. 122px,517px), t indicates the time of the first point in the fixation group (e.g. 1275ms or frame 1275), and d indicates the total duration of the fixations in the group (i.e (timestamp of the final classified fixation - timestamp of the first classified fixation)(2125 - 1275 = 850 milliseconds, the tota duration of this fixation group) [2].

Due to the algorithm's simplicity, it only requires one parameter to be specified, the velocity threshold. This threshold is usually computed using angular velocity, but this is only possible if the distance between the subject's eyes and screen are known. Without this, certain aspects of the data collection need to be approximated in order for the threshold to be inferred. This collection generally involves exploratory data analysis and assumptions, such as the sampling frequency being constant, thus abandoning angular velocity in favour of a vector-based velocity calculation formula [2]. This is what will be used during the implementation of this algorithm for the time being.

A clear benefit of this algorithm is in it's simplicity, and it is likely to function to a reasonable extent in many different eye-tracking tests. Despite this, the results may also be inconclusive and may require additional algorithms to reinforce the results.

Identification via Hidden Markov Model

The second velocity-based algorithm is Identification via Hidden Markov model or I-HMM. As mentioned above, this algorithm builds on the basic properties of I-VT, albeit with some additional functions. A unique aspect of the Hidden Markov Model algorithm is that it relies on a probabilistic approach when it comes to distinguishing between features and saccades. Rather than using velocities thresholds from the data at any given point to explicitly classify a fixation or saccade, it iteratively uses points at an earlier state combined with previous event classifications to predict if forthcoming points shall be classified as high-speed saccades or low-speed fixations. The probabilistic aspect of the Hidden Markov Model is that it initially builds from the basic velocity-based approach, but is reinforced by two additional algorithms. The first algorithm goes over existing classifications of fixations and saccades and re-classifies them by maximizing a likelihood function. This is done taking into consideration not only the velocity of each individual point but also its neighbours, such as the current state the classifications are in, as well as the probability of certain points belonging to one event, which can help with identifying transition periods between events. The second additional algorithm then simply updates the existing points with the probability parameters determined by the first algorithm, with the intention of creating a more accurate result building upon the initial I-VT approach [19].

Based on the probabilistic nature of this approach, it is used in certain human-computer interaction applications such as handwriting or speech recognition. The potentially more accurate results of the I-HMM approach are due to its non-fixed nature, allowing the velocity threshold to be more flexible depending on the nature of the experiment or situational eye-movement [2].

As explained above, the I-HMM algorithm is actually three algorithms and can be classified as a velocity-based algorithm since the baseline for the Hidden Markov Model is I-VT. Therefore, some of the technical details of this algorithm can be found in the I-VT technical description (**??**) on page **??**. The probabilistic parameters from above can be described as a two-state algorithm, where each state refers either to fixations or to saccades. With this, there will exist two probability states, one for the observed probability that one point is either a fixation or a saccade, and the other that determines the probability of a transition from one state to another. Since probability ranges from 0 to 1 (where 0 is the absence of any likelihood and 1 is certainty), the likelihood of a point being a fixation will fall between these two values, with the remaining probability summation falling into the saccadic category (e.g. if a point has a fixation probability of 0.75 or 75%, the remaining probability of the point being a saccade will be 0.25 or 25%). The I-HMM algorithm will predict what points will be fixations or saccades using dynamic programming. If the velocity of points is increasing and the probability skew is shifting from one state to another, the algorithm can predict that a transition period is in effect. For this observation, the algorithm uses two parameters ("mean and variance of the distribution") for the purposes of observing the state [2]. Additionally, the algorithm can be tweaked to include angle as a parameter as well.

The whole algorithm in a basic sense works as follows. First the point-to-point velocities are calculated in a set. A decoding process then determines the maximum likelihood of each given point (based on the previous velocity calculation) and thus identifies certain points as either fixations or saccades (despite the values not being binary) [2]. This decoding reclassification process is usually done using the Viterbi sampler. The sampler is based on the algorithm of the same name, designed for solving "the problem of estimating the state sequence of a discrete-time finite-state Markov process observed in memory-less noise", which in this specific case is based on the probabilistic parameters [25]. The re-estimation process is then undertaken to re-evaluate the parameters and set them in place. This is normally done using the Baum-Welch re-estimation algorithm. As this is a recurring process for re-evaluating the parameters, the original algorithm is a "technique which occurs in employing the maximum likelihood method in statistical estimation for probabilistic functions of Markov chains" [26], and will additionally minimize errors in the classification. For effective results, several re-estimations will occur for any given data [27]. The fixation groups are collected, with saccade groups omitted (similar to I-VT). Finally, each fixation group is mapped to the centroids of their respective groupings, and then returned as results [2].

While complex, re-estimation can be used for the I-HMM parameters. This technique utilizes training data to learn the specific parameters and can be used for any future eye-tracking tests occurring on the same setup with the same protocols [2].

While more complex than the basic I-VT algorithm, the potential of more accurate results is a consideration for this algorithm.

Besides the more well-known velocity based algorithms discussed above, there exist more contextual, focused algorithms that can still work effectively for eye-tracking data outside their original experiments.

Binocular-Individual Threshold

A more complex algorithm involving velocity based methods is the B-IT or Binocular-Individual Threshold. This algorithm specialises in differentiating small saccades from noise, which can be seen as a form of tackling the significant problem of data loss. It does this by looking at the movements of both the left and right eye's simultaneously, constantly monitoring if they are doing the same behavior at the same time. For example, with other algorithms, if the left-eye gaze data is noisy in a certain time-span of the experiment while the right-eye data is mostly clean, the algorithm will interpret the data incorrectly, and can cause miss-classification and loss of accuracy. B-IT on the other hand, will look at the data differently, and determine that the noisy eye is roughly at the same place as the other eye. This accountancy for noise, while checking each eye's data, can help determine if the peaks in velocity are real, or just due to noisy data. This being said, this is a more advanced algorithm that also uses an adaptive threshold for classification [19].

Van der Lans et al. in 2010 claims that the algorithm improved upon existing velocity-based algorithms in three ways. The first feature has already been explained, the usage of binocular viewing (both eyes accounted for in classification), using covariations between the two eye's movement information to identify and classify between fixations and saccades. The second feature is that it starts by estimating the velocity-threshold, rather than using a fixed threshold. It was already mentioned previously that the algorithm using such an adaptive threshold, but it seemingly works automatically. There is no need to define a threshold prior, and the algorithm adjusting from it as it automatically finds the starting threshold. This estimation allows the adaptive threshold to account for different eye-movement directions as well as varying participants and their respective tasks in an experiment. Lastly, "it accommodates the inherent stochasticity in eye movements", essentially the randomness/noise or data as described prior for the proposes of not incorrectly classifying certain data points that cross over the velocity-threshold [28].

Nyström & Holmqvist / Larsson, Nyström & Stridh

Another algorithm is the Nyström & Holmqvist algorithm, which was introduced earlier in 2.6.1 and 2.4. It is notable for it's ability to identify post-saccadic oscillations (PSOs, called "glissades" in the paper [10]) while simultaneously identifying fixations from saccades like the basic I-VT algorithm. It is however not a fixed velocity threshold, adapting itself based on how noisy the data provided is [19]. This algorithm should provide more accurate results, as it will not erroneously identify PSOs as saccades, rather as the beginning of fixations.

An iterative approach to the above velocity threshold algorithm is the Larsson, Nyström & Stridh algorithm. This 2013 algorithm is essentially a further development of the 2010 Nyström & Holmqvist algorithm, but the algorithm is now able to detect saccades separately. What truly makes this algorithm a significant advancement over previous work is that it has to ability to detect said PSOs and saccades within the presence of smooth pursuits. This is particularly challenging since smooth pursuits are not usually classified as fixations or saccades as they present a problem for fixed velocity thresholds [19]. From these additional factors alone, the expected outcome of this algorithm should be even better and provide even better results.

2.6.2 Dispersion/Displacement-Based Algorithms

Another common approach that is also intuitive, but perhaps slightly more complex in its implementation are dispersion or displacement-based algorithms. These algorithms take advantage of the dispersion of fixation points, which is the distance the points are spread across from each other. This algorithm assumes the fixation points occur close to one another. Naturally, an approach like this is duration sensitive, as fixations are based on how long the points remain in the dispersion area [2]. The challenge here is to determine what spread is ideal as one is more likely to capture false fixations if the spread threshold is too high. In addition, a specified duration threshold must be determined for differentiating saccades and fixations within, or just outside a specified area. Many different algorithms exist using said methodology, and will be discussed below.

Identification via Dispersion Threshold

The first dispersion-based algorithm is identification via dispersion threshold or I-DT. Like with the I-VT algorithm, it is also a very popular approach to eye-tracking feature separation, being a relatively simplistic algorithm for which more advanced techniques would later build from. Instead of the features being separated by the rate of change of position, they are now explicitly being separated solely by their positions relative to each other, including the behavior in only where they move, instead of how they move. Like with the I-VT algorithm, this places an emphasis on fixation identification, disregarding saccadic periods.

The basic idea of the algorithm is that it classifies points based on their two-dimensional values. It has two fixed thresholds. First is a "minimum fixation duration threshold", meaning that the points must remain for a set amount of time to be classified as fixation points. The second is the "maximum fixation dispersion threshold", meaning that the points must be in a defined area to be classified as fixation points. Any points fulfilling both of these threshold are then considered parts of a fixation [19].

Like with the I-VT, this algorithm is not too complicated relative to the more advanced velocity and dispersion algorithms discussed in this section. As stated, this algorithm takes both the spatial (proximity of the points) and time related (accumulated duration of the points) aspects of the data into account [27]. When fixations are found, they will contain a centroid with the corresponding diameter from the maximum fixation dispersion threshold [2].

The valued aspect of the algorithm can be defined as a temporal window, where certain eye-movement actions must take place within a certain time frame for a general classification, in this case, against set proximity threshold. The time span within this window can be considered as a stopwatch, and when combined with a spatial value, determines whether or not the points inside this entire threshold can be considered fixation points. If said points to not meet this threshold, the temporal window is then moved and restarted at the next point the resides in it's respective spatial proximity, and the process is then repeated. The previous point space is then classified as a saccade. The maximum dispersion threshold is calculated using the formula $D = [max(X) - min(X)] + [max(Y) - min(Y)]$ with X representing the horizontal relative positions and Y representing the vertical positions of the points. In our case, this will refer to the pixel-positions on the screen, but normally this would again be done with a visual angle (like with the originally proposed I-VT algorithm). This is combined with the minimum fixation duration threshold that will vary depending on the experiment or eye-tracking equipment used (e.g. 100ms, mainly since fixation durations are typically at least this long [2]) [27].

The algorithm in a procedural sense works as follows. First, a spatial window will be "opened" as long as there are enough points for classification. Once enough points have been covered to satisfy the minimum duration threshold, the conditional statement for the minimum spatial threshold will then be conducted. If the dispersion of points fits within the minimum threshold, said points will be added to the temporal window (classified as fixation points) until the points are no longer within the threshold. A fixation point is then noted as the centroid of the fixation for that duration. The windows points are then removed from the recorded points for that fixation period. Any points falling outside the conditional statement are disregarded until a given point falls inside a new threshold once again. The fixations are then returned at after the end of the recorded points [2].

It is easy to see why this algorithm is considered robust as it uses clear boundaries for classification. While a solid stepping stone, it can certainly be improved upon with some extra parameters or algorithms to improve not just the accuracy, but precision of the results.

Covariance via Fixation Dispersion Threshold (F-DT/C-DT)

The second dispersion-based algorithm is covariance via fixation dispersion threshold or C-DT. This is an approach derived Veneri et.al's earlier works on the basic fixation dispersion threshold or F-DT. The original F-DT algorithm is based on the I-DT methods of clustering due to its robust nature. The key difference is that a two-sample F-test is used for the purpose of "equal variances to evaluate variance around the centroid". In a sense, it aims to adjust the centroids of the original I-DT to be more accurate. A critical objective of the algorithm to making sure the x and y boundaries are not significantly different from one another. The F-test used is the same as a "null hypothesis", aimed at verifying "that two populations have the same variance". H0 (null hypothesis) is to be tested against H1 for the heterogeneous variances [29].

The C-DT algorithm developed from the F-DT algorithm incorporates covariance calculations into the x and y gaze data. These extra calculations helps with the violation-sensitive nature of F-tests when it comes to "normality assumption of the data", thus variance, co-variance and duration thresholds were incorporated into the algorithm to account for potential outliers and inaccuracies [19].

The procedure for this algorithm is certainly more complex than the original I-DT algorithm as it incorporates F-tests for the purpose of an even more robust algorithm that produces better results. The first task undertaken is to identify the centroid of a fixation. It starts using the same proximity calculations like the I-DT, but proceeds to calculate the F-test on the given X and Y gaze-points at that specific timestamp. It will then do this for all the further given points as long as it is within the fixation proximity. Once the fixation area is finished, the centroid is then determined through the F-tests conducted earlier for variance purposes. The fixation is then extended, accounting for the centroid point. The "ratio of variances" for the X and Y is then calculated within the fixations. A process is then conducted to reduce the ratio of variances by extending the lower fixation limit. The same will happen for gaze points at "timestamp + 1", where the upper fixation limit is extended instead again for the purposes of reducing the ratio of variances. Once all the fixation points are gone through, the centroids are then applied to said features at their respective timestamps and the new fixations are then returned [29].

The algorithm appears very complex, perhaps too complex to be implemented successfully consistently. Perhaps it may only work with a very large amount of data to get accurate variance scores for the F-tests to work as intended. While the potential for the algorithm is high, it might not work as well as one can expect.

Identification via Minimum Spanning Tree

The third dispersion-based algorithm is identification via minimum spanning tree or I-MST. As can be inferred from the name of the algorithm, it makes use of trees that obtain the eye-tracking data, where each branch of a tree is a unique data sample. It utilizes "samples from two different clusters" and it does this by capturing data that only has as much branching as the threshold requires. These clusters are then captured by two separate nodes from higher up in the whole tree instead of unnecessarily branching out to a very low-level node. The idea here is to classify saccades primarily, so they can first be excluded from any fixation detection. It does this by "enforcing certain thresholds on the samples at the edges of a cluster" [19]. The main idea is to minimize the lengths of the line segments in the tree, hence minimum spanning tree [2]. These lengths can be defined as the euclidean distance "among all spanning trees in a given set of nodes". The I-MST algorithm builds the MST necessary for classification, which does so, based on the dispersion/point-to-point distance thresholds [27].

The trees described earlier are constructed using Prim's algorithm. It works by initializing a tree with just 1 vertex and no arcs. For each interval (looping through the sample) "it adds to the tree an arc of minimum weight connecting a vertex in the tree with a vertex outside of the tree". This is done until the spanning tree is created [30]. Using this method, there should exist one MST for every sample (set of points). This method better defines "subsequent characterizations of defined fixations" compared to the previous fixation filtering methods [2].

The actual I-MST algorithm requires evaluating the already generated MSTs for identifying fixations. As a first step, the maximum depths of the interconnected branches are discovered. Branches that are considered to be too low (close to the edges) in the MST are considered inadequate for separating fixations. The separation of edge lengths are considered and compared with the use of the mean μ and standard deviation σ of said lengths. Using this, a framework in generated. It controls "where fixations may occur" in any given MST within an experiment. It also determines "how local adaptivity affects fixations decisions" [2]. The power of MSTs allows for additional fixation characterization parameters, such as critical paths, which are areas in an MST that have "minimal branching structure" [2].

The algorithm works as follows. First, an MST is constructed from the experiment eye-gaze data using Prim's algorithm. Next, the maximum depth of the branches are found for every point (node) in the MST. Third, the saccades are then classified as edges that exceed the depth threshold. Fourth, mean μ and standard deviation σ parameter values of the edges are defined. Edge lengths over a set threshold ratio are also classified as saccades. The fixations are then defined as "clusters of points not separated by saccades". Said fixations are then returned [27].

Said algorithm seems fairly complex, and thus difficult to understand. It however promises good results using complex methods for classification. It may also prove difficult to implement and the process might take time due to its complexity as a system. If the results prove outstanding, implementing this algorithm may have value.

2.6.3 Area-Based Algorithms

A less common, but still widely known approach are area-based algorithms. These algorithms are a little different as they attempt to identify points of interest by using "relevant visual targets" rather than basing the classification from gaze-point to gaze-point. Fixations at any specified areas of interest are provided by "lower-level identification and high-level assignment". Fixations can be used as input for area-based algorithms as well as represent relatively "higher levels of attentional focus" on any port of a viewing space, and can thus be classified as "macro-fixations" [2]. This being said, it is clear this approach is more advanced that velocity or dispersion-based approaches. While there exists many proprietary algorithms in different studies, only one well-known algorithm exists on a general scale.

Identification via Area of Interest

The main area-based algorithm is identification via area of interest or I-AOI. The area of interest algorithm works significantly differently from the other algorithms and approaches discussed prior. While the previous methods are able to identify fixations at any point inside the eye viewing space, this method relies on identifying fixations that occur at specific target areas inside the visual space. Thereby, it assumes where fixations should occur inside the field. These specified regions are defined rectangles (based on the two-dimensional X and Y coordinate space) and "represent units of information" within the visual space. These defined rectangles are designed to identify fixations close to relevant, informative targets. In order to distinguish passing saccadic points from prolonged fixation gazes inside the coordinate space, the algorithm also utilizes a duration threshold, similar to the dispersion-based algorithms starting on page 20 [2].

The algorithm finds the precise target areas using associated data points, thus labelling these points as fixations, and classifying everything outside these areas as saccades, regardless if certain gaze-data outside of these areas has fixative behavior. If the points in a certain target area do not meet the minimum duration threshold, the whole fixation group is discarded. The remaining fixation groups are then turned into fixation tuples. More details of the algorithm will be explained in the next paragraph [2].

The algorithm works as follows. First, every point within the vicinity of a target area is labelled as a fixation point, and every point outside of a target area labelled as a saccade. Afterwards, consecutive fixation points are collapsed into typical fixation groups, classifying any stray or sparse points inside a target as saccade points and therefore removing them. Once these fixation groups are generated, they are checked if they meet the minimum duration threshold. Those groups that do not meet this condition are then removed. Now that the fixation groups are mostly finalized, the centroid points within the target areas are created for each of the fixation groups. These groups are then returned as fixations [2].

It is difficult to tell from the outside if such an algorithm will work well. More contextualized experiments and perhaps conducting predictive eye-movement tests can make this approach more interesting. But for general eye-tacking experiments with less visual information, it can be expected to lose accuracy.

2.6.4 Combined Algorithms

The approaches below contain algorithms considered to be less common than what is usually used to filter eye-tracking data. Nonetheless, they can prove novel and perhaps perform better in certain areas, with the potential trade-off being that they are perhaps harder to understand and therefore implement due to their more advanced nature.

Identification via Kalman Filter

The first combined algorithm is identification via Kalman filter or I-KF. This algorithm combines the elements from both the velocity-based and positional-based algorithms discussed in-detail earlier. This algorithm is a recursive (repeating) filter aimed at improving precision and minimizing error via combining current and predicted measurement from previous input values via gaze-data [19]. Since the algorithm uses both velocity and position for classification, it is thus a two-state system with the two respective properties. Combined with acceleration, as well as "applied to the recorded eye position signal", the predicted eye-gaze velocities are generated with the algorithm (I-KF) [27].

The algorithm using this filter (I-KF) does not classify the eye-data into events like all of the other have so far, rather it is done using a X^2-test. The algorithm classifies fixations of this test value is below a certain threshold while simultaneously fulfilling a minimum duration threshold. Conversely, if points are found to be above this threshold, they will be classified as saccades, regardless of the duration threshold set. Like with prior algorithms, the generated fixations are grouped via near-proximity clustering methods [19]. The χ^2-test is also known as the Chi-square test and is written as:

$$\chi^2 = \sum_{\mathrm{i}=1}^{\mathrm{p}} \frac{(\hat{\theta}_{\mathrm{i}}^{-} - (\dot{\theta}_{\mathrm{i}})^2}{\delta^2} \tag{2.1}$$

Where $\hat{\theta}_{\mathrm{i}}^{-}$ is the predicted eye velocity and $\dot{\theta}_{\mathrm{i}}$ is the actual observed velocity from the eye-tracker itself, δ indicates the standard deviation of the each velocity, p indicates the size of the "temporal sampling window" and finally X^2 is the calculated threshold for which gaze-points will are either determined to be fixations or saccades [27].

Once again, the algorithm appears to be complex and may not be implemented in this thesis. However, due to its combined nature via the two-state system, combined with a predictive generated results makes this a very powerful algorithm. It combines the robustness of the previous methods, along with a focus towards improving the precision of eye-gaze classification while reducing error.

Chapter 3

Data description and methodology

3.1 Data collection

The data used in this thesis was collected from a previous experiment conducted by Samip Bhurtel and co-workers. This experiment partially involved the use of an eye-tracker to collect raw data of the users' eye movements as apart of communication training "towards non-verbal people". The data collected from the physical experiments is relevant to this thesis, as the data contains fixations and saccades to the nature of the experiment (users observing letters and numbers for communicative purposes) [24].

As apart of the ethical application in the thesis, the data was collected in Norway. A requirement for publishing a study involving human research is that it must comply with the NSD (Norsk senter for forskningsdata). The participants' information as well as their involvement with the research is considered sensitive data, and therefore must be considered confidential.

Figure with illustration of saccades and fixations, separately (or different color). Using the separation algorithm from the eye-tracking software that collects and separates data, we can plot a basic scatter of the relative fixations and saccades.

The data in this section was taken from a fraction (10%) of a single participant in the experiment. Of the 4655 data points captured, 3670 were classified as fixations, 569 were classified as saccades, and the remaining 416 were unclassified, and were thus not used.

In the figure 3.1, rough estimations for the fixations and saccades are highlighted, saccades paths are shown as blue line and fixations as red blobs. While the data is not perfectly accurate from the eye-tracker, it gives a solid idea of the visual differences between the two features. Fixations are clearly marked out, while some points have seemingly been erroneously

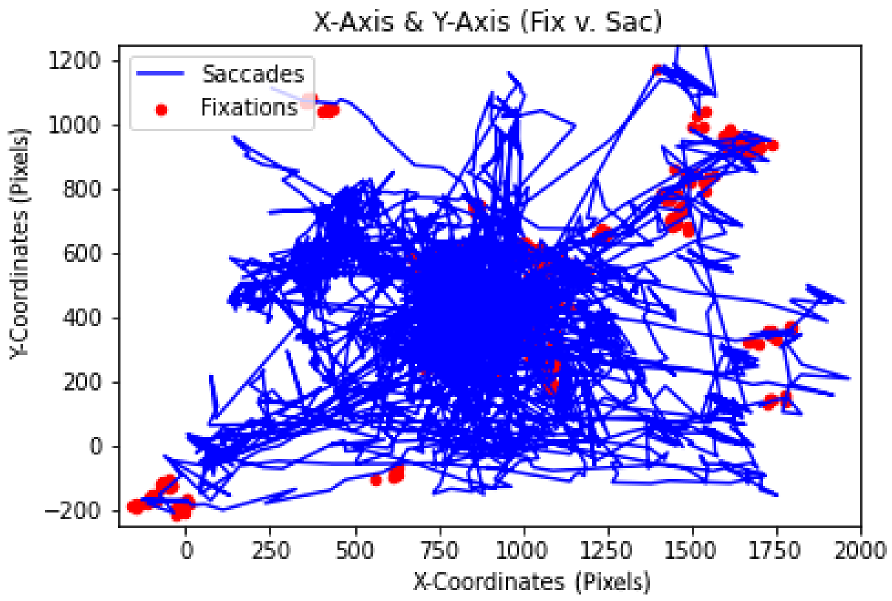


Figure 3.1: Fixations and saccades compared, with approximated saccade paths.

defined as saccades, despite clearly belonging fixation areas along the visual space. This is of course only speculation through visual observation.

3.2 Quantitative characterization of fixations and saccades

Using the algorithm from the software to distinguish between fixations and saccade, plot for each group separately:

1. Angle (θ) as a function of duration (t). The angle is the degree separation between 3 points, where the middle point acts as the pivot for the angle to be calculated. Using internal (dot) product, we can calculate the inner angle (θ) for each saccade point, as well as each fixation point, using the following formula:

$$\theta = \cos^{-1} \frac{(x \cdot y)}{|x||y|} \tag{3.1}$$

   Where θ is the calculated angle and x and y are the lengths of the two adjacent vectors forming angle θ. The angles between each vector for the whole data set will be calculated for the purpose of forming comparison between the average angles of fixations and saccades.

   The results for the angle calculations with respect to duration are shown in figures 3.2 and 3.3 for fixations and saccades respectively. The results for both features seem to reflect opposite visualizations,

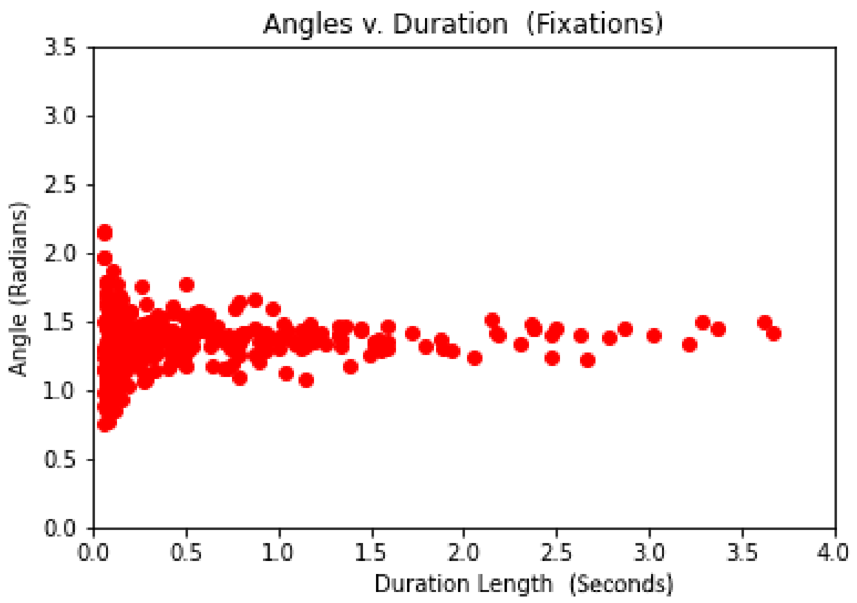


Figure 3.2: Graph for the angles of fixations with respect to duration.

which possibly indicates relatively accurate results since both features are classified for opposing reasons. While the fixation graph shows a mostly horizontal path for the scatter, the saccade graphs show a mostly vertical path for it's scatter.

While several fixations and saccades both last for very short periods of time (under 0.25 seconds), many fixations seem to last for considerably longer periods of time, up to 3.5 seconds, while the vast majority of saccades last under half a second (0.5 seconds). In addition, the duration of the features appears inversely proportional to the variation (spread) of the average directional change in the features. The angle for each feature represents an average directional change per point within that specific feature (for example, if 5 points were classified as one saccade, the average of the 4 directional changes will be the angle result in the above graph). The lower the duration length of a feature is, the higher chance it will deviate from $\pi/2$ (90 degrees), which indicates less circular eye-motion. Saccades deviate from this value as eye-movements during saccadic periods either remain travelling in a relatively straight line (closer to 0 radians) or "turn around" towards an overshot fixation area (closer to π radians or 180 degrees).

2. Velocity (v) as a function of corresponding duration (t) of the fixation or saccade. Velocity is essentially the speed of an object in a certain direction, thus making it a vector. The most basic calculation of velocity is computing the average velocity, where the difference of distance over the difference of time results in a velocity value.

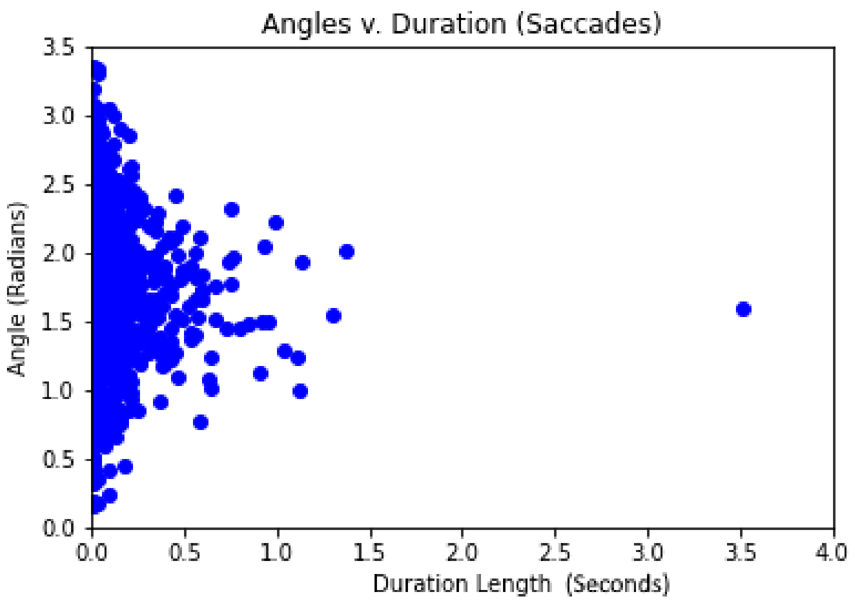


Figure 3.3: Graph for the angles of saccades with respect to duration.

$$\bar{v} = \frac{\Delta x}{\Delta t} \tag{3.2}$$

Where Δx is the distance between two points and Δt is the time elapsed between two points. Alternatively, the equation can be stated like this:

$$\bar{v} = \frac{x_1 - x_0}{t_1 - t_0} \tag{3.3}$$

Where x_1 and x_0 are the start and end points in the vector, t_1 and t_0 are the start and end of the elapsed time, and v is the calculated velocity. In this case, the displacement $(x_1 - x_0)$ between two points is calculated using Pythagorean theorem, as the points are along a two-dimensional space.

Since the eye-tracker calculates the eye movements in a point-to-point nature, it is acceptable to assume one trajectory for each calculation, therefore not having to be concerned over the direction of the vector, strictly for this calculation. Like with the angular calculations above, the velocities for both saccades and fixations will be separated for the purposes of comparison.

The results for the velocity calculations are shown in figures 3.4 and 3.5. As noted above, the fixation feature duration tends to be considerably longer than for saccadic features. This corresponds with the resulting velocity figures, where again, velocity and duration

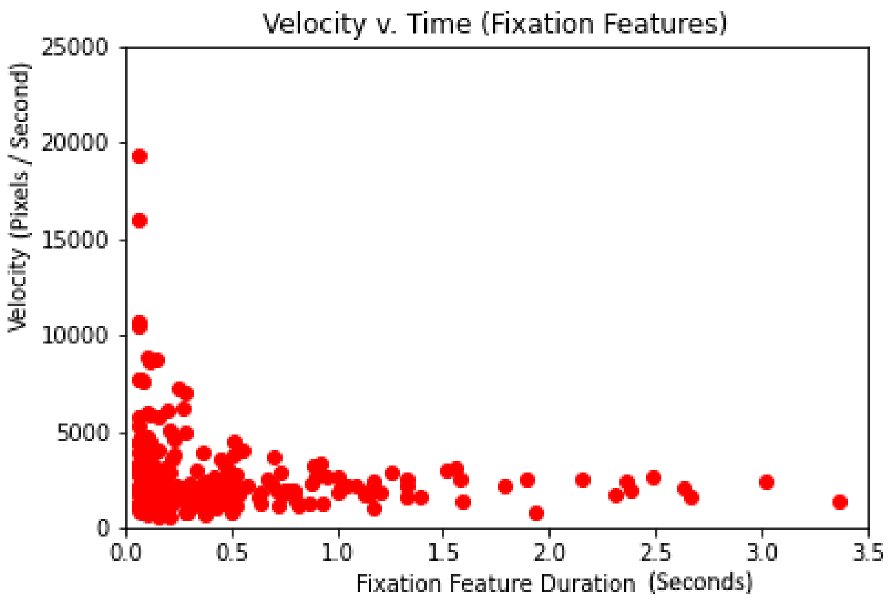


Figure 3.4: Graph for the velocities of fixations with respect to duration.

results are mostly inversely proportional (higher feature duration correspond with lower average velocity). This includes the mostly opposing results of fixations and saccades, where higher average velocities (>5000 pixels per second) for saccades appear much more common.

The main observation, is while fixation durations last longer than saccade durations, the average speed of the fixations remains relatively low compared to saccades. This is most likely due to the fact that fixation movement occurs in a confined area resulting in smaller and therefore slower eye movements. Saccades are the opposite, as they are movement-heavy transitional states between fixations, and therefore last a shorter period of time, travelling across a relatively long distance at a much higher average speed.

3. Acceleration (a) as a function of time (t). Acceleration is most simply defined as the rate of change of velocity, meaning that the acceleration of an object (e.g. human eyes) is just the derivative of velocity with respect to time, therefore $f'(v) = a$.

This can be written as:

$$a = \frac{dv}{dt} \sim \frac{\Delta v}{\Delta t} = \frac{v_1 - v_0}{t_1 - t_0} \tag{3.4}$$

Where a is the acceleration between two vectors, Δv is the velocity difference between the start (end of the first vector) and end of the second vector ($v_1 - v_0$), and Δt is the difference in time between

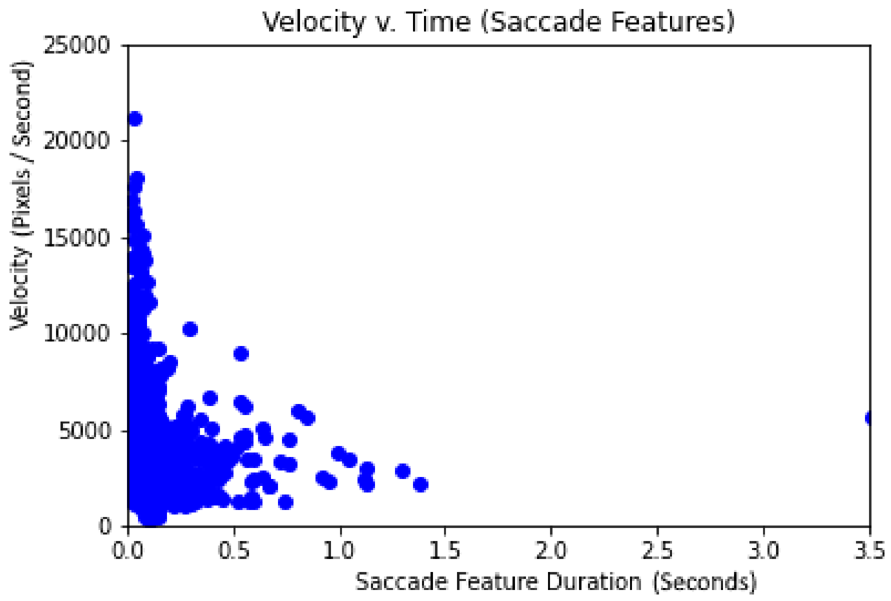


Figure 3.5: Graph for the velocities of saccades with respect to duration.

the two points in the vector ($t_1 - t_0$). The saccade and fixation acceleration results will be separated as with the angle values and velocity calculations above.

The acceleration results for both fixation groups and saccades groups are demonstrated in figures 3.6 and 3.7 respectively. Like with the previous two comparisons with respect to duration (feature duration), the fixation and saccade feature results contrast each other (and are once again, inversely proportional), where fixation accelerations seem to vary much less than saccadic accelerations. Naturally, the acceleration spread between acceleration (>0 pixels per second) and deceleration (<0 pixels per second) remains mostly uniform in both cases since the eye will naturally speed up, and slow back down when moving either regardless of feature classification.

Fixation values seem to not vary much relative to saccades. While fixation accelerations never go beyond or below $0.25x10^6$ (<250000 pixels per second), saccadic accelerations occasionally go beyond $0.50x10^6$ (>500000 pixels per second). This lack of fixation acceleration is due to the relatively constant speed (but not velocity, due to change in direction) of eye-movements within a fixation. As a consequence of the small area fixations generally occur in, there is little opportunity for rapid eye-movements, thus the speed differences between detected points in a given fixation are going to be minor, and therefore only equally small acceleration values.

Saccadic acceleration values tend to vary greatly due to the transitional period from a fixation to saccade (the main accelera-

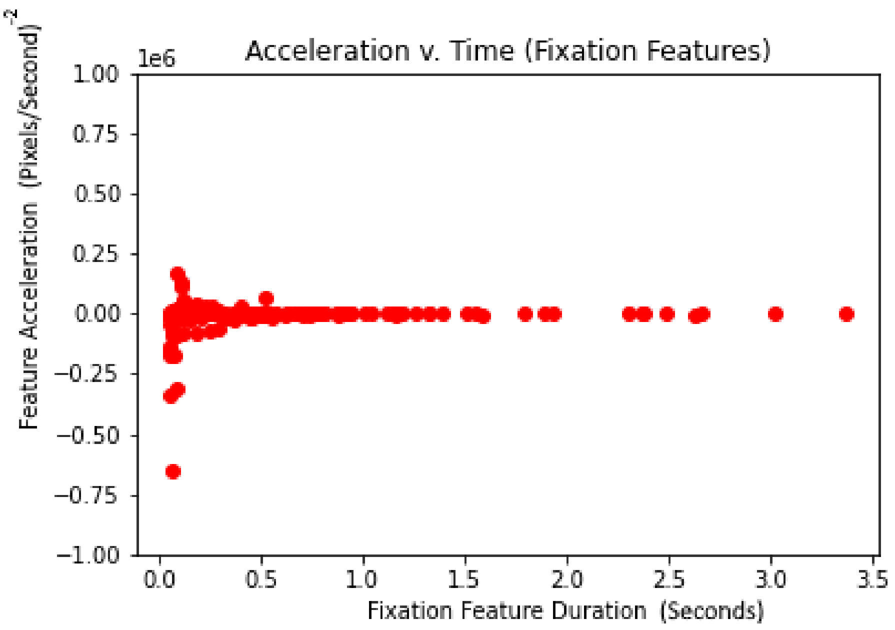


Figure 3.6: Graph for the acceleration of fixations with respect to duration.

tion/deceleration period) are still classified as fixations, outliers in the saccadic data are most likely anomalies in the recording. It can be assumed that due to the high acceleration values of fixations, the velocities and distance travelled relative to time will be high, and therefore these periods will simply not last as long as fixation periods.

4. Scatter plot of $\theta \times v$, $\theta \times a$, $a \times v$.

   The results for the comparisons between the angles and their respective average velocities are shown in figure 3.8. What can be observed instantly, is that the saccades are more scattered compared to the fixations with regard to both the average angle changes and the average velocities from point-to-point. What is also clear is that average velocity does not seem to affect average angle much. This is clearly seen by the mostly circular shape in **??** where high/low angle changes co-exist with high/low average velocities within any given saccadic feature. In both cases however, velocity values on the outer ends of the scatter plots seem to have average angle changes between 1 and 2 radians, which coincides with the angle/duration comparisons in figures 3.2 and 3.3 from earlier.

   The results for the comparisons between the angles and their respective average acceleration values are shown in figure 3.9. Once again, it can be observed that the saccadic results are more spread compared to the relatively concentrated area of the fixation scatter. As seen earlier from 3.6 and 3.7, the acceleration results for saccades were much more spread out due the nature of those specific eye movements. As with above, it appears that the average angle does

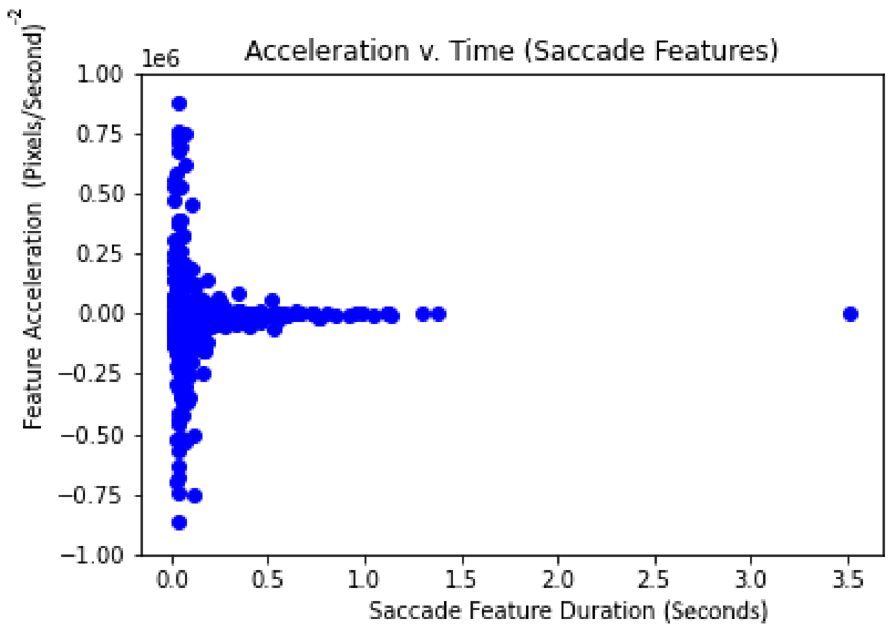


Figure 3.7: Graph for the acceleration of saccades with respect to duration.

not affect the acceleration or deceleration of the eye. This may be because the eyes during saccadic periods will inevitably accelerate and decelerate as a part of the transition process from/to a fixation, regardless of which direction the eyes may turn during this period. The same can be said for fixations, where the angle in which the point-to-point eye movement happen are independent of the relatively little acceleration the eyes inevitably experience during a fixation period.

The results for the comparisons between the velocities and their respective average acceleration values are shown in figure 3.10. What one may expect is that low average velocities may coincide with low acceleration values, but once more, the values appear to not directly affect each other in either case. In the case of saccades, high acceleration saccadic periods (>1000000 pixels per second) can occur when at average velocity periods as low as 5000 or as high than 10000 pixels per second. Like with the previous saccadic results, the points lie in a mostly circular spread. In the case of fixations, the results are once again concentrated, with consequently lower average velocity and substantially lower average acceleration results. The results between the two scatters are not proportional. While saccadic velocity results appear two or three times higher than fixations, saccadic acceleration results appear several times high or lower in comparison.

5. Histogram of the features' angles. Now that the angle θ of each point between the vectors has been plotted against the function of time (t), the angles themselves can be plotted as a histogram on their own. The

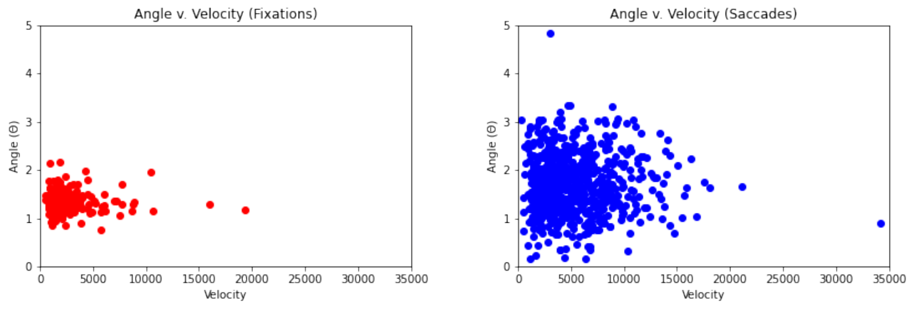


Figure 3.8: Scatter plots comparing angles and velocities of the fixation features, and comparing angles and velocities of the saccade features. Qualitatively, we do not see any significant correlations.

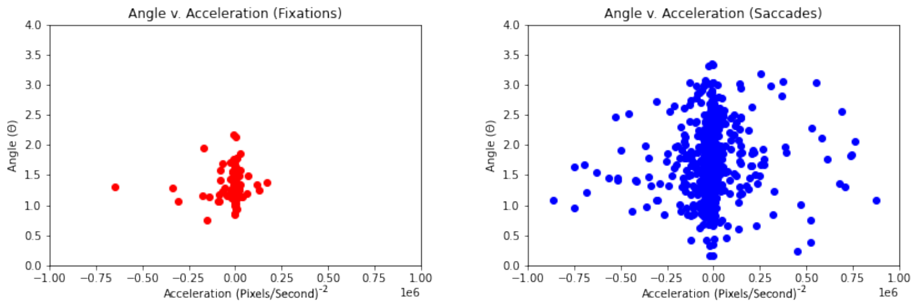


Figure 3.9: Scatter plot comparing angles and accelerations of the fixations features.

histogram values for both fixation angles and saccade angles will be plotted on a single graph. A hypothesis can be made that the values for both will be distinguishable even on a single graph.

The resulting histograms for the angle changes from point to point, as well as average angles from the features are illustrated in figure 3.11 and 3.12 respectively. In the case of figure 3.11, the fixations and saccades have been combined for better visual observation. Normally, just the fixation and saccade groups would be exclusively presented here, however in this case the individual vectors' angle changes will be included and focused on. Using grouped results requires the implementation of the average angle for each vector in a given grouping. This results in relatively inaccurate resulting angle groups that do not correlate at all with the individual values presented here. This can especially be seen when comparing the individual saccades and saccade features.

When comparing the individual fixation angles to the saccade angles, a clear contrast can be seen. As expected from the fixations, the majority of the results are closer to π, which indicates the eyes "circling" the fixation point, occasionally re-correcting or simply fluctuating in-place. Meanwhile with saccades, the biggest group of

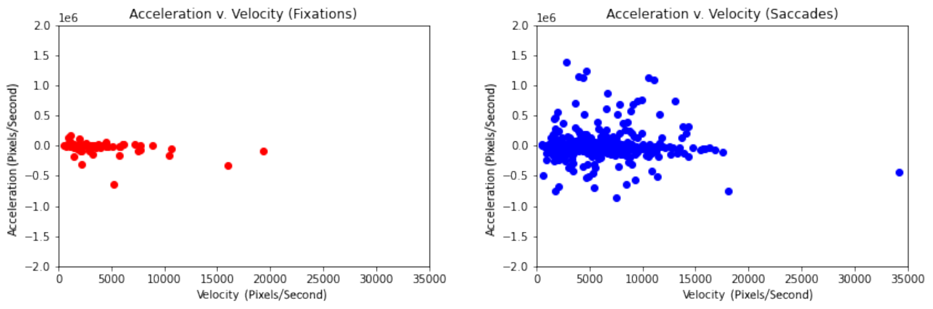


Figure 3.10: Scatter plots comparing accelerations and velocities of the fixation features, as well as the saccade features.

points are closer to 0, indicating a small degree of angle change. This is because saccades are mostly straight-line travel from one visual space to another. There are inevitably direction changes (e.g. sudden direction changes to a different fixation than originally intended or a "scanning", ziz-zag like behavioral pattern en-route to a fixation area). The other notable observation are the large amount of points closer to π. This is most likely due to both overshoots (resulting "u-turns" from overshooting fixation areas) as well as saccadic periods, where the eyes leave a fixation area for a brief moment, and return back. The nature of both can be seen clearly by the fact that the average angle change for fixation and saccades are 1.77 radians (more than 90 degrees) 1.49 rad (less than 90 degrees) respectively, indicating movements in fixations closer to the origin points, and further away during saccades.

6. Histogram of the duration of the features. Additionally, it would also be interesting to see how long the duration is for fixation and saccadic periods, as a way to compare the density of the points and perhaps infer the occurrence of each feature and how many are to occur over a determined amount of total time. Like before, the histograms will be separated for better comparison clarity.

   The resulting histograms for the feature duration of both the fixations and saccades are seen in figures 3.13. The clear observation here are the more varying results for fixation periods. While short fixations still exists, there are many cases where fixations last longer than saccades. This is the expected result as fixations are when the eye-focuses on an object, taking into account the human subject processing the visual information. Saccades are just subconscious eye-movements going from one place to another. These movement usually coincide with high velocities, and therefore these periods only last short times. In this case, the average saccade period is just 0.144 seconds, while the average fixation period is several times higher at 0.633 seconds.

   It is worth noting that fixations periods can sometimes be short. This

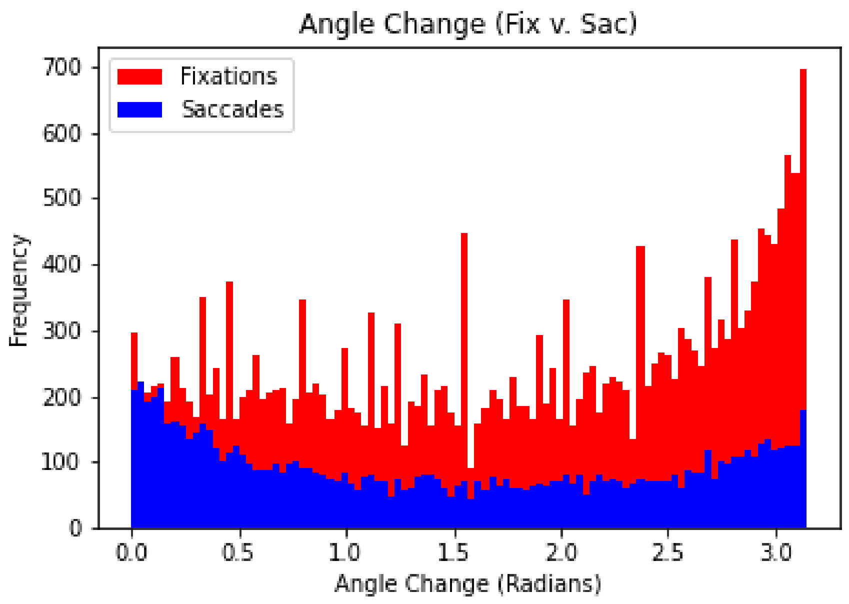


Figure 3.11: Histogram of the angles, comparing the fixation and saccade points.

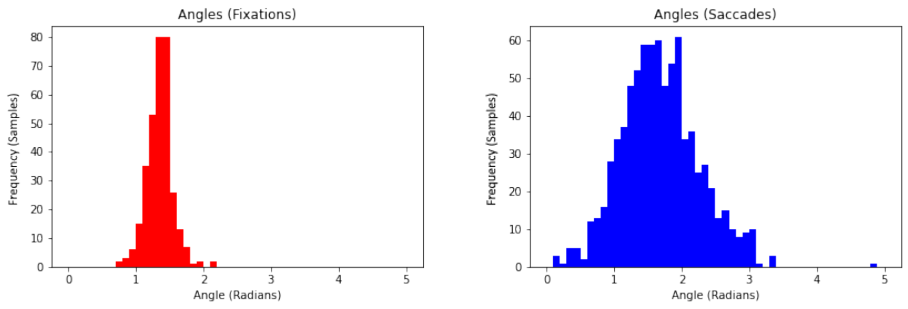
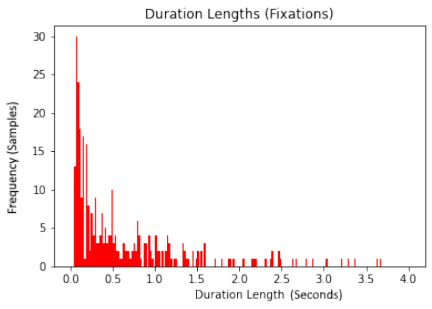


Figure 3.12: Histograms of the angles of the fixation and saccade features.

could either be mis-classification, where fixation groups are briefly split by unusually short saccadic periods. Sometimes fixations may simply not need to last long, and the user may have accidentally observed a wrong visual element, correcting to where they originally intended to look.

7. Histogram of the features' velocities. Like with the angles plot above, making another plot specifically for velocity should also prove useful. Like with the angular plot, the velocities for both saccades and fixations should vary significantly from each other, granting a clear picture of the nature of both features. This will also be plotted on a single graph due to the hypothesised disparity in the values.

   The resulting histograms for both the feature and single vector velocities of the fixations and saccades are seen in figures 3.14 and



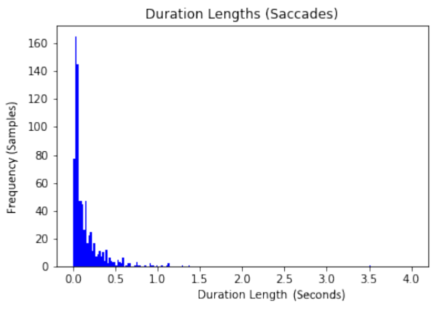


Figure 3.13: Histograms of the angles of the fixation and saccade feature durations.

3.15 respectively.

When comparing the four graphs, it can be seen that the saccadic gaze-velocities are significantly higher than in fixations. The average fixation point velocity is around 1986 pixels per second, while the average saccade point velocity is around 5365 pixels per second, more than 2.5 times faster. This is to be expected as saccades are just travel zones between fixation gazes. This can be seen more clearly in 3.14, as the velocities in the saccade graph have a higher spread towards right, indicating higher velocities, meanwhile a large portion of the fixation velocity values remain lower and more concentrated on the graph. It is important to note that some saccade have low resulting velocities. This could be down so simply error in the data, or that some fixations are erroneously classified as saccades as they may have either just entered, or exited a fixation. The same phenomenon can be seen for fixations with higher-than-average resultant velocities. The main takeaway is the visual observation that the majority of the fixation point are concentrated closer to zero, while the largest concentration of saccade values hover around 5000 pixels per second.

The calculated feature velocity shows similar results. Just from visual observation, the similarities for groups and points can be seen, where the saccade group is shifted more towards the right, indicating higher velocities. The numerical results are similar too, with fixation and saccade features averaging 1923.74693694 and 4615.64392586 pixels per second respectively. Both results are lower than their point velocities, it is unknown why.

8. Histogram of the features' acceleration values. While a little more obtuse than the angles and velocity for results plotting, acceleration can still prove useful to visualize. This should provide additional detail on the differences between saccades and fixations. The plotting of the graph will follow the same nature from above.

   The resulting histograms for both the feature and single vector

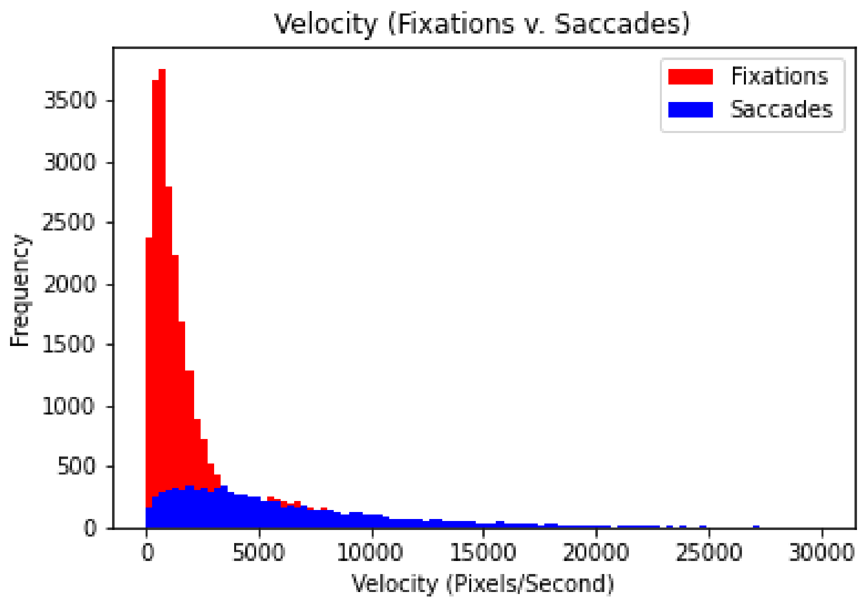


Figure 3.14: Histogram for the velocities for each of the determined fixation and saccade points.

accelerations of the fixations and saccades are seen in figures 3.16 and 3.17 respectively. The single vector results are interesting. For the fixations, only one spike in values can be observed, meanwhile for saccades, two spikes are observed. This is most likely because acceleration and deceleration periods are apart of whole saccadic periods. For the majority of the time, only a relatively small amount of acceleration occurs as the eye adjust around an area at a mostly constant, low speed. The two spikes being asymmetrical in the saccade graph could just be down to error, as one would expect mostly similar amount of acceleration compared to deceleration since the eye speeds up from fixation speeds, and goes back down to those speeds after a saccade. From a numerical point of view, the average fixation acceleration is as-expected, with a low number of -752.571 pixels per second. The saccadic results are vastly different, with a high average result of 134862.304 pixels per second. This high value may simply be down to error. What is also interesting is that the result contrasts the graph. The larger spike occurs in the eye-deceleration as opposed to acceleration, yet the resulting average does not echo the same observation.

The feature results have been partially excluded since the resulting overage accelerations were distorted by outlier results and thus provide meaningless information. These results also do not showcase the major differences in the two types of eye-movements as averaging both accelerations should result in close to zero. Therefore, there is little in the way of comparison, other than noting the variance in the

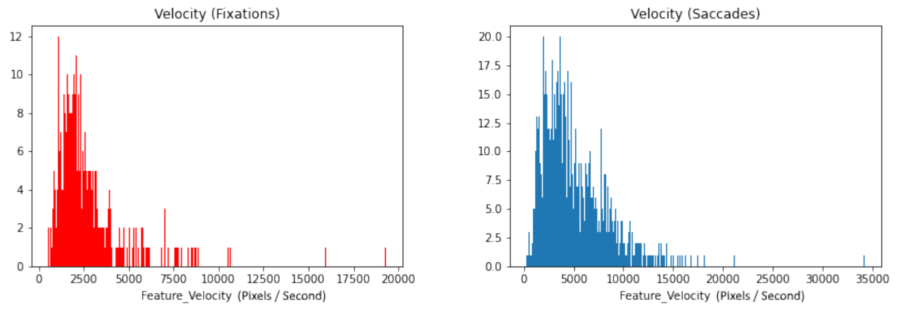


Figure 3.15: Histograms for the velocities for each of the determined fixation and saccade groups.

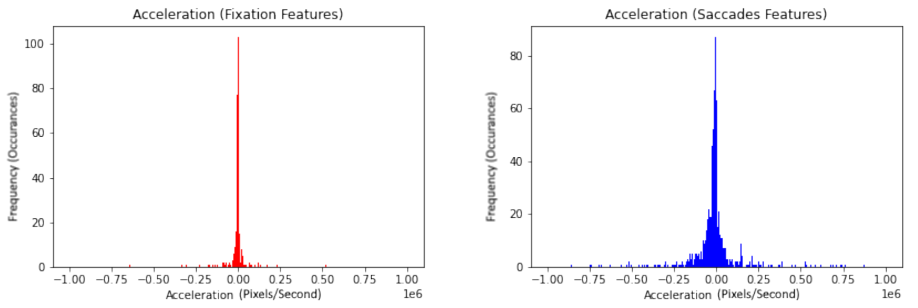


Figure 3.16: Histogram of the accelerations for both the fixation and saccade features.

saccadic results are higher than in fixations.

3.3 Methodology & Implementation

A few of the prominent algorithms for eye-tracking classification.

1. Identification via Velocity Threshold (I-VT)
2. Identification via Hidden Markov Model (I-HMM)
3. Identification via Dispersion Threshold (I-DT)

The algorithms selected for implementation are listed above. These algorithms will be used for the purposes of classifying fixations and saccades while omitting data that is considered not apart of either. These point classifications will then be grouped into fixation and saccadic features. Consideration of results, as well as implementation difficulty were taken into account. All implementation is done in Python 3 using JuPyTer Notebook. The notebooks for both the data analysis done above 3 as well as the implementation described here can be found the main GitHub page [31]. For the purposes of readability, the code will not be shown in this section, but rather described in an easy-to-understand way.

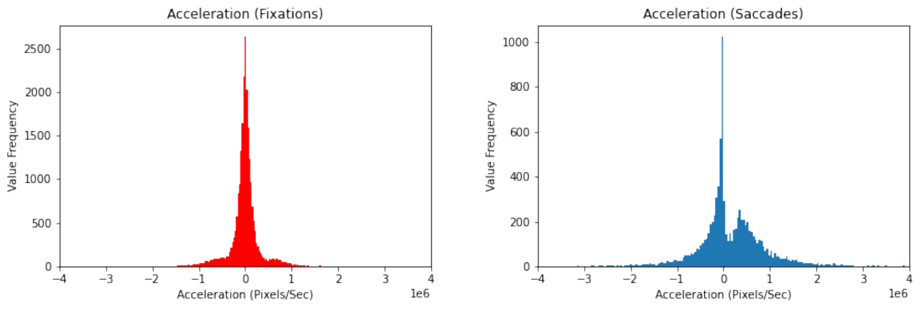


Figure 3.17: Histogram of the accelerations for both the fixation and saccade points.

3.3.1 Identification via Velocity Threshold (I-VT)

Implementing the I-VT algorithm requires just one end parameter, velocity, specifically the velocities of the vectors that connect two eye-gaze points in secession. For this, there are two additional parameters that must be retrieved, the euclidean vector distance (or the displacement of two succeeding points) and the time taken to get from one point to the other.

Calculating the euclidean vector distance (displacement) is done using traditional Pythagoras ($\sqrt{a^2 + b^2} = c$) where a and b are x-lengths and y-lengths respectively. These are calculated simply by subtracting the first point's coordinates from the second point's coordinates. Once these values are retrieved, the c value from the Pythagorean theorem can then be calculated. Any calculations with missing x or y coordinate data are discarded, as the result will not be a number.

Calculating the time difference is much simpler. This is done by looking at the timestamp for both points and finding the difference (subtracting the first from the second) between them. Like above, any points with missing time data are discarded. Additionally, any points where the time difference is zero are also discarded. This is because these values will produce errors later on (diving by zero gives NaN (not a number) values).

Every valid velocity vector will then be assigned the resultant velocity magnitude and all classified vectors will be done so based on the "end" point of that vector (thus the first point in the whole series is excluded). The velocity threshold is user defined. For example, if the decided velocity threshold is 500 pixels/second, any vectors exceeding that velocity are classified as "saccades", while any vectors that are less than, or equal to the threshold, are classified as "fixations". For all classifications, their respective displacement, time difference, angular difference (described in 3) and timestamp and X/Y coordinate values are added to new lists alongside the fixation/saccade classification list for further analysis.

3.3.2 Identification via Angle-Velocity Threshold (I-AVT)

As an alternative to I-VT, the I-AVT was conceptualized. This was a novel idea created during the thesis with the main goal of improving the simplistic velocity threshold algorithm. As the name suggests, this is an algorithm that iterates on top of the original existing I-VT algorithm. This algorithm simply incorporates a portion of the angle calculation from equation 3.d in section 3.2 and applies the cosine to the resultant angle. This calculation is applied to each corresponding vector's velocity (Vector velocity V_2, where V_1 and V_2 are the first and second vectors' velocities respectively required to calculate the angle change):

$$V_{eff} = V_2 * \cos\theta \tag{3.5}$$

Where V_{eff} is the new I-AVT result and $\cos(\theta)$ is the cosine of the resultant angle of the two velocities of their respective vectors. The idea behind this algorithm is that the angle should have an effect on the original velocity and reflect a more relatively accurate velocity based on the previous velocity in a series of points. Additionally, angles that are closer to zero or π will have a lesser impacted result on the velocity, which should help better classify saccades, as they should be closer to those values since their angle changes should be lower. This includes fixations as they have more varied, unpredictable angle changes, thus "penalizing" high velocity, extreme (closer to $\frac{\pi}{2}$) angles within probable fixation groups. This should also help filter singular or outliers points for both features.

3.3.3 Identification via Dispersion Threshold (I-DT)

Implementing the I-DT algorithm was a more precise and complex process, involving more conditions and parameters for proper classification. Besides the usual parameters from before like X/Y coordinate position, there also exist temporary parameters that act as a "pending" status for the values to be classified either as fixations or saccades. For this implementation, there are temporary lists for possible X/Y fixation coordinates as well as their respective timestamps. Additionally, these X/Y coordinates are added to temporary centroid lists for further centroid calculation (if said points are eventually classified as fixations). The main final parameters are the X/Y calculated centroid coordinates, the classified fixation/duration points, and the duration of each fixation.

Before going through all the points in the data, a start point is defined as the very first point in the series. This start point indicates the beginning of the first possible fixation period. Along with this, the X/Y coordinates for this start point are defined. The timestamp for this point is considered the start of the duration period, and the duration difference for the next possible fixation takes this first point as the start. The end points are defined exactly the same like the start points, the key difference being that the end point is the current data for which the pointer inside the loop is located. In other words, while a fixation is being "determined", the end point is the current point in the loop through the whole amount of data.

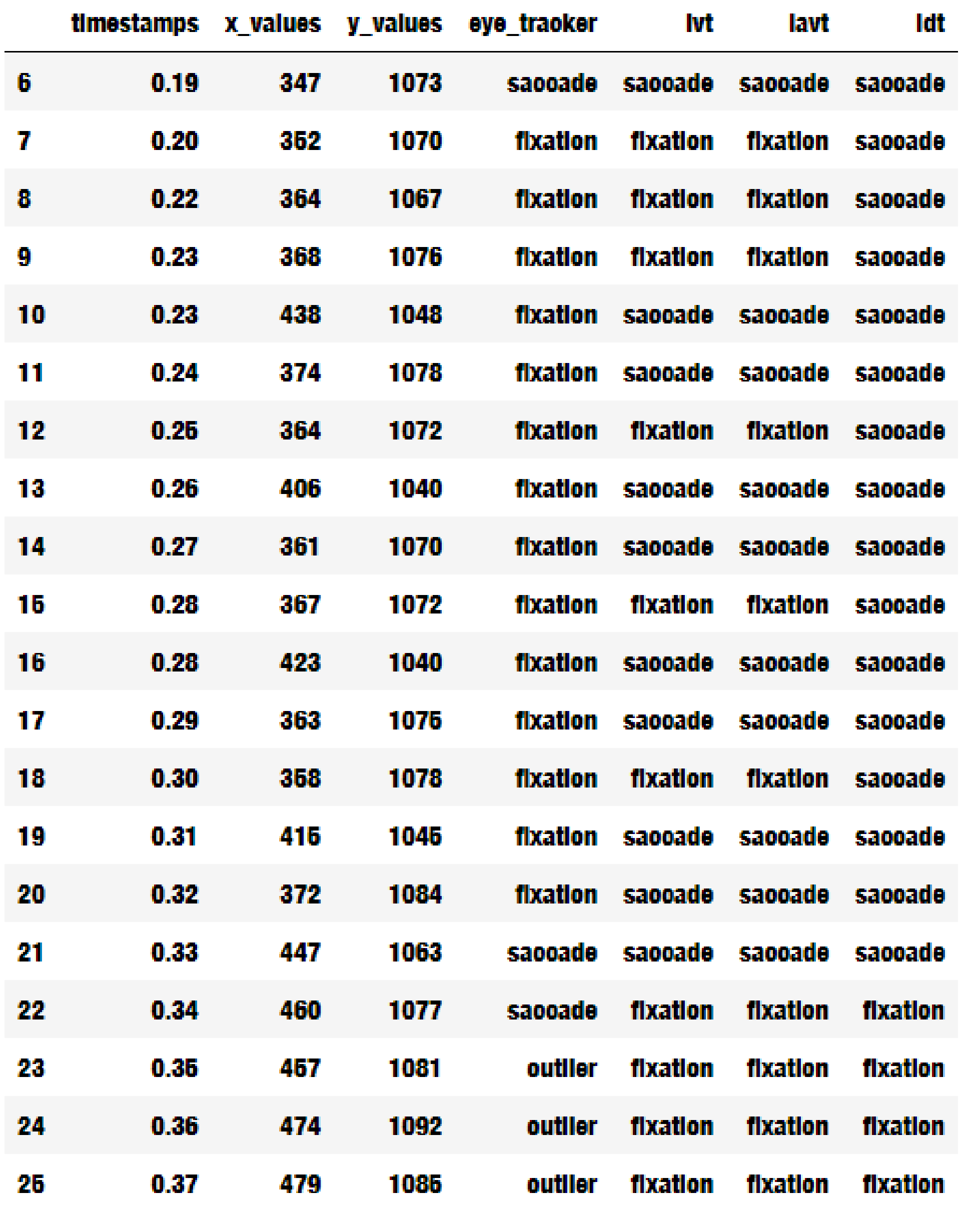
With these two pieces of information established, the differences can be established for classification purposes. The duration (time difference) of a fixation is simply defined as $(\textit{end point time} - \textit{start point time})$, where the *start point time* is the timestamp for the first point in that fixation periods, and the *end point time* is the current point's timestamp within the same fixation period. This gives the information for how long the current fixation is running for, and is compared directly to the duration threshold. This duration threshold is user-defined, and must be fulfilled for a series of possible fixation points to be grouped as a fixation. For fixation dispersion conditions, the X and Y coordinate boundaries must be established, this is simply done as $(|\textit{end point } x - \textit{start point } x|)$ and $(|\textit{end point } y - \textit{start point } y|)$. Where the *start point* is the defined coordinate point for the start of that specific fixation period, and the *end point* is the current point's coordinates for that same fixation period. These differences must each not exceed the defined boundaries from the start point for them to be classified as fixations.

The first conditional checks if the current Y and X coordinates are within the set proximity (for example, 200 pixels) of the start point. If it is met, it then checks if the duration threshold is met. If this threshold is met. The X/Y coordinates, timestamp are put into fixation lists, and the set is classified as a fixation. Additionally, they are also put into the temporary centroid lists for calculation at the end of that fixation group. The fixation duration is then defined at this point (which can increase with further valid fixation points in the same set). If it is not met, the "would be" fixation points, and timestamps are put into temporary fixation lists. Like with the points above, they are also put into temporary centroid lists.

If this first condition is not met, it will check if fixation being evaluated meets the defined fixation duration (0.2 seconds for example). If it does so, all of the points and timestamps in the temporary lists are then put into fixation lists and are properly classified as fixations. The centroids are then calculated using the average positions of the X/Y point coordinates for that fixations. These are then added to a list alongside a separate list that adds the duration of that respective fixation. All of the temporary points are then reset, and the start point is now defined as the current point immediately after the fixation has ended. This current point is classified as a saccade to split the previous and fourth coming fixations.

However, if the duration is not met after the fixation group, the coordinates and timestamps inside the temporary lists are put into the saccade lists and are all properly classified as saccades. The current point is again defined as a saccade like with the previous conditional statement. Like with above, the temporary parameters are all cleared, the possible centroid values are discarded and the start point is defined as the current point in the loop, for a possible new fixation period.

3.4 Data Table



	timestamps	x_values	y_values	eye_tracker	ivt	iavt	idt
6	0.19	347	1073	saccade	saccade	saccade	saccade
7	0.20	352	1070	fixation	fixation	fixation	saccade
8	0.22	364	1067	fixation	fixation	fixation	saccade
9	0.23	368	1076	fixation	fixation	fixation	saccade
10	0.23	438	1048	fixation	saccade	saccade	saccade
11	0.24	374	1078	fixation	saccade	saccade	saccade
12	0.25	364	1072	fixation	fixation	fixation	saccade
13	0.26	406	1040	fixation	saccade	saccade	saccade
14	0.27	361	1070	fixation	saccade	saccade	saccade
15	0.28	367	1072	fixation	fixation	fixation	saccade
16	0.28	423	1040	fixation	saccade	saccade	saccade
17	0.29	363	1075	fixation	saccade	saccade	saccade
18	0.30	358	1078	fixation	fixation	fixation	saccade
19	0.31	415	1045	fixation	saccade	saccade	saccade
20	0.32	372	1084	fixation	saccade	saccade	saccade
21	0.33	447	1063	saccade	saccade	saccade	saccade
22	0.34	460	1077	saccade	fixation	fixation	fixation
23	0.35	457	1081	outlier	fixation	fixation	fixation
24	0.36	474	1092	outlier	fixation	fixation	fixation
25	0.37	479	1085	outlier	fixation	fixation	fixation

Figure 3.18: A table with a sample of the classification results of the implemented algorithms.

The figure 3.18 is a sample of the classification results (from points 6 to 25) from the eye-tracker classifier as well as the implemented algorithms. It includes x-axis and y-axis data as well as the corresponding timestamps for those points. One of the results determined from this table is the hamming distance, which is a value that describes the discrepancy (difference) between two series of values (or lists). In this instance, the hamming distance will be computed for all the implemented algorithms, comparing their classification to the original eye-tracker's classification algorithm to determine how different the results are from each other and to also determine how well that particular algorithm performed. Additionally, this will be done through the use of typical classification performance parameters such as precision and recall. These more detailed results of the algorithms are presented and discussed in the next chapter.

3.5 Data processing & Feature Grouping

The data processing and grouping of the features is done using an algorithm that takes the resulting fixation and saccade list and groups them based on consecutive results. The condition for a new group is that there must exist at least two consecutive classifications from the same feature. Once a new feature is pointed to, a group is created from the previous points. These conditions do not apply for outlier classifications, thus they are always non-grouped and exist individually. The groups (as well as individual points) are put into a new 2-dimensional list, where individual points and groups exist in the first dimension, and the contents of the groups exists in the second dimension. This grouping is relevant to the next feature that is discussed below.

3.6 The K-Ratio

What makes this approach to these algorithms novel, is the inclusion of our K-Ratio algorithm. The main idea behind this ratio, is that it helps with categorizing which of these algorithms works best. It does this by finding out the best value (the lower the K value, the better) for each applied algorithm to help each algorithm achieve a hypothetical best possible result. Therefore, this should be a reliable addition to the table further in the results that compares the implemented algorithms.

First, we separate our sample into two groups A and B (e.g. fixations and saccades). If the probability to transition from one group to another is independent of the current group, then the expected transition rate from A to B (and vice versa) is:

$$p_{independent A \to B} = n_a \times (1 - n_a), \tag{3.6}$$

Here, n_a is defined as N_a/N, where N_a is the number of occurrences in group A and N the total number of occurrences.

If our groups are separate, we should observe that the k-ratio is defined as

$$k - ratio = \frac{p_{empirical A \to B}}{p_{independent A \to B}} \tag{3.7}$$

Where the k-ratio should be lower than one.

$p_{empirical A \to B}$ is just the actual observed transitions between the defined groups in the classification (e.g. how many times the classification transitions from A to B and vice versa.)

While this criteria can prove useful in our case as saccadic and fixational movements tend to be sequential across time, it does not guarantee completeness. For example, if a consecutive set of saccadic displacements is grouped with fixational movements, the ratio k should not be altered.

Chapter 4

Results

4.1 The K-Ratio

The K-Ratio was applied to two of the three algorithms implemented in this thesis, the I-VT and the I-AVT. Thousands of increments were carried out to display the K-Ratio curves, as well as for observing the best possible results. The implementation was done using the amount of saccades over the whole data set for probability P. In the case of I-VT, the 4 instances of outlier classifications were included in the $p_empirical$ formula as their own groups as excluding them will not have a significant effect on the results (8 of approximately 9000 transitions). The K-ratio was not implemented for the I-DT algorithm due to the parameters, which is discussed in section 4.6.

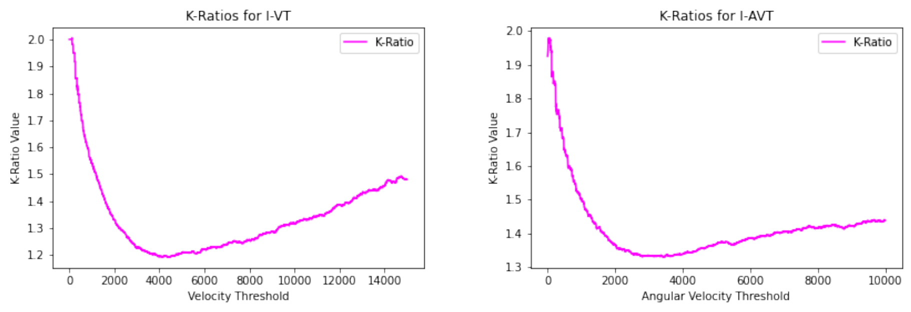


Figure 4.1: Plots of the K-Ratio results for the I-VT and I-AVT algorithms.

Above in 4.1 are the K-Ratio results for the I-VT and I-AVT algorithms across thousands of tested thresholds. Along the X-Axis are the velocity and angle-velocity thresholds respectively for classifying fixations and saccades within their own algorithms using the sole parameter. Along the Y-Axis is the K-Ratio, where the lower the ratio is, the better the result based on the expected probability. What can be observed, are the curves that both graphs produce, where an ideal point is clear on both of them and the results become gradually worse the further away from the ideal point the thresholds are.

For the I-VT tests, the best K-Ratio value achieved was a 1.193 at a velocity threshold of 4368 pixels per second. Further, the best result for the I-AVT was a 1.330 at 3462 relative pixels per second. As can be observed, neither result is below 1, which is not ideal. This will be investigated further down in section 4.6.

4.2 The I-VT Algorithm

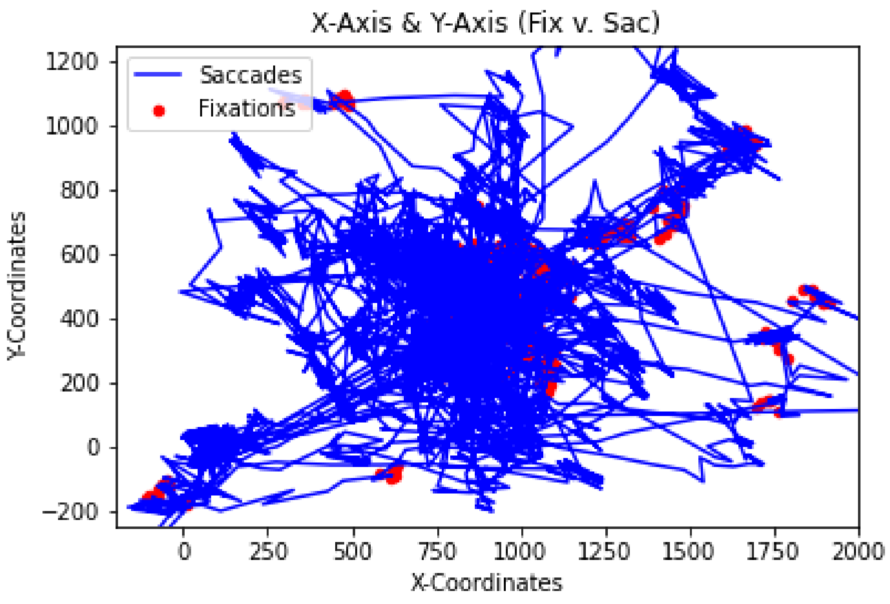


Figure 4.2: Plot showing the fixation and saccade gaze-points separately on the visual space, using the I-VT algorithm

The results of the fixation and saccade gaze-points for the I-VT can be seen in figure 4.2. This results shows the visual space of the X and Y range in the test. It is worth noting that the results here are only taken from a sample of the data set for better visual clarity. The fixation points are marked as red spots while the saccades are shown as blue paths. Using a low velocity threshold of 2000 pixels/sec, the fixations and saccades are relatively clear. There are obvious fixations point in places, as well as logical saccade paths connecting said fixations.

The results for the fixation and saccade velocity comparisons for the I-VT can be seen in figure 4.3. Here the velocity threshold split is clear. The red area marks the velocities of fixation points, while the blue area marks the velocities of the saccade point. As stated previously, the split occurs at 2000 pixels second, which is just above the average fixation velocity (1986.05125866 pixels per second) from the original eye-tracker algorithm.

The results for the displacement comparisons between the fixations and saccades can be seen in figure 4.4. The most immediate aspect to note

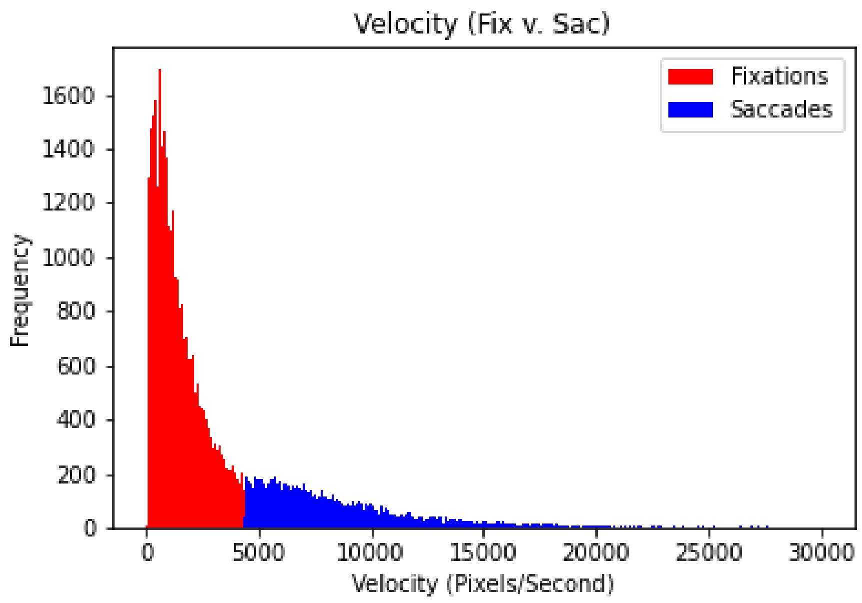


Figure 4.3: Histogram of the velocities comparisons for both the fixation and saccade points, for the I-VT.

is the very similar trajectory of the point occurrences compared to the velocity histogram. The peak is at a similar point, as well as the uneven curve towards the right (with a hump occurring at around 50 pixels). This is inline with the threshold results as well, with the split occurring at roughly 20 pixels. Interestingly, there is almost no overlap between the two classifications in the graph.

The results of the scatter plots, comparing the fixations and saccade points with displacement and velocity can be seen in figure 4.5. This graph compares the displacement with the velocity for both classifications. The additional lines with higher slopes represent gaps in the timestamps. Additionally, the velocity threshold can be seen here, where minimal overlap is shown with displacement, as one can see by the sparsity of the points beyond the lines in the zoomed figure of 4.5. The takeaway is that the results are simply limited by the quality of the data, as a significant portion of the are isolated points with no data on continuity due to missing parameters (e.g. x/y-axis or timestamp data). This data discrepancy makes it more difficult to justify their classifications.

The histogram comparing the angle changes for the fixation and saccade points can be seen in figure 4.6. Comparing the results of fixations and saccades, there is a clear divide in the angle changes between the two features. The fixation angles remain rather uniform with slight biases towards forward movement (0 radians) and "U-turns" (3.14 radians). On the other hand, saccades have clear trajectories, with extremes at 0 radians and 3.14 radians. This is reinforced by visually inspecting 4.2 as most

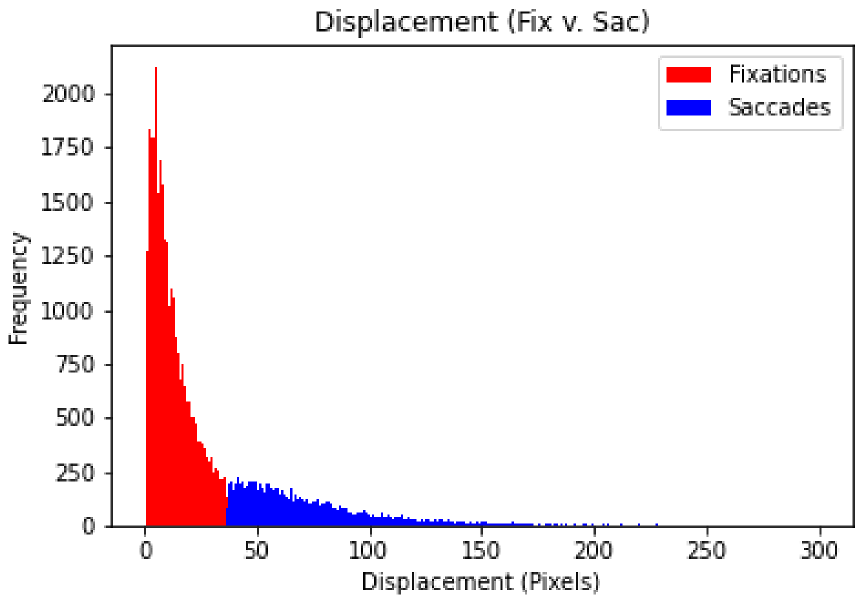


Figure 4.4: Histogram of the displacement comparisons for both the fixation and saccade points, for the I-VT.

saccadic behavior either show forward trajectory (with natural occasional deviations) or extreme movements in the opposite direction. For this result, the I-VT seems quite effective, showcasing a clear difference between fixation and saccade eye-gaze behavior.

4.3 The I-AVT Algorithm

The results the gaze-plots containing the fixation and saccade movements for the I-AVT are found in figure 4.7. Using the same reduced data-set, one can inspect the classification of fixation and saccades with the red fixation points and blue saccade paths. What is clear is that the classification is skewed... When comparing to the standard I-VT from 4.2 the results are different as there are more saccade paths as opposed to fixation points.

The histogram comparing the algorithm threshold results for both the fixations and saccades are found in figure 4.8. This graphs shows the histogram for the specific I-AVT calculations described in 3.3.2. Naturally, new values will now fall between both positive and negative due to the cosine injection. What can be seen is a relatively large amount of values that fall closer to zero. Otherwise, through visual inspection, a mostly even distribution result can be observed, with a split between positive and negative values. It can be inferred that most values that are closer to zero (less than 3500 (greater than -3500) relative pixels per second) can be classified as fixations, as shown here. The split between the red and blue sections indicates this classification. As can be seen in the angle figure

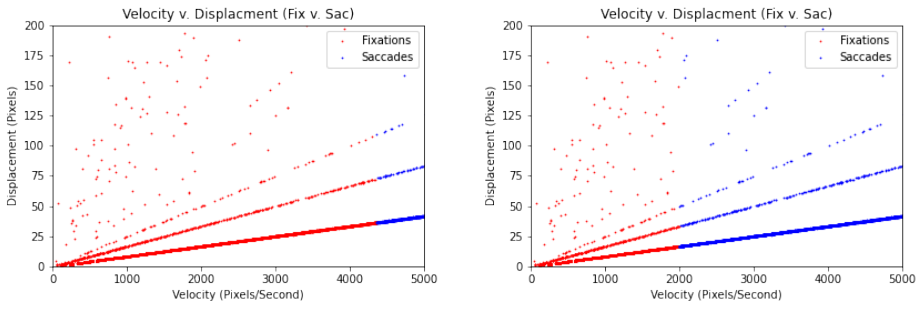


Figure 4.5: Scatter plots comparing both the fixation and saccade points in relation to displacement and velocity simultaneously (unmagnified left, magnified right), for the I-VT.

(4.10) further below, a far greater amount of fixations compared to saccades will be affected by the cosine "penalty". From this, a far greater amount of fixations will be classified compared to saccades, which is considered normal due to the nature of each eye movement feature.

The histogram containing the velocity comparisons for both the fixation and saccade values are contained in figure 4.9. This graph shows the strict velocity data for the classified fixations and saccades. It is the exact same velocity data that is present in the I-VT results in 4.3, however it is distributed differently due to the updated algorithm. With that in mind, it is interesting to compare the two results. The most obvious difference is that there is an overlap in data, meaning certain velocities generally classified as saccades may appear apart of the fixation grouping and vice-versa. Despite this, there is a clear velocity group for both features, and no fixations are as fast as the average saccade, and the same applies for saccades when compared to average fixation velocities. A takeaway is that the I-AVT algorithm stays true to the velocity threshold aspect of the algorithm, while simply iterating on top of it.

The results for the new angle parameter in the velocity threshold are seen in figure 4.10. As this is the new parameter that transforms the standard I-VT into the I-AVT, it was important to see how the new resulting classification would look like when directly impacting the results of the distribution. When comparing these to the I-VT classification in 4.6, one can observe a more distinct classification, where the saccade values have shifted more towards the boundaries of the angle range, while almost no points remain in the middle. Interestingly through visual inspection, fixation classification has remained mostly the same, seemingly unaffected by the shifted saccade classification. Thus, the fixation result is similar, where the angle changes are relatively evenly distributed across the range. What this does prove is that adding angle change as a parameter has had a significant impact on the classification results. Based on the inference that saccades should exhibit directional change behavior that is either minimal or extreme, drives the fact that this classification is more accurate than the

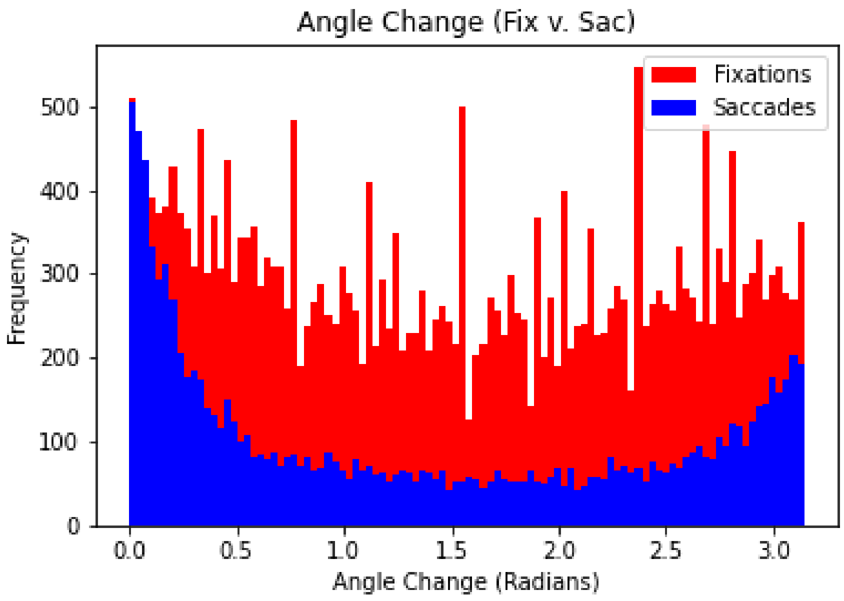


Figure 4.6: Histogram of the angle change comparisons for both the fixation and saccade points, for the I-VT

results from I-VT.

4.4 The I-DT Algorithm

Like with the previous algorithm, figure 4.11 contains the eye-gaze data for the I-DT specifically. This figure shows how this specific I-DT configuration classifies fixations and saccades from the visual space. Like with, I-VT, this result is only the partial data set. It is clear that while some fixations are classified properly, some are also erroneously classified as saccades (dense "blobs" of short blue paths). This can probably be attributed to the low proximity threshold as opposed to a low duration threshold.

Figure 4.12 contains the resulting centroids for the I-DT algorithm, corresponding to the set thresholds. While the velocity-based I-VT also contains centroid calculations for the fixation groups, it was not relevant as the I-VT results proved more visually effective using the singular points only, thus grouping and centroid calculation were ignored. This shows the fixation centroids for every classified fixation grouping. This result also included the reduced data set to correspond with the results in 4.11. These centroids were calculated using the average X and Y-axis positions for all the points in that respective fixation. As can be visualised, the fixation centroids and the classified fixation points correspond with each other, where the rough centers of the classified fixations can be seen clearly.

The histogram containing the duration comparisons between fixations and saccade features are found in figure 4.13. Since the algorithm uses duration

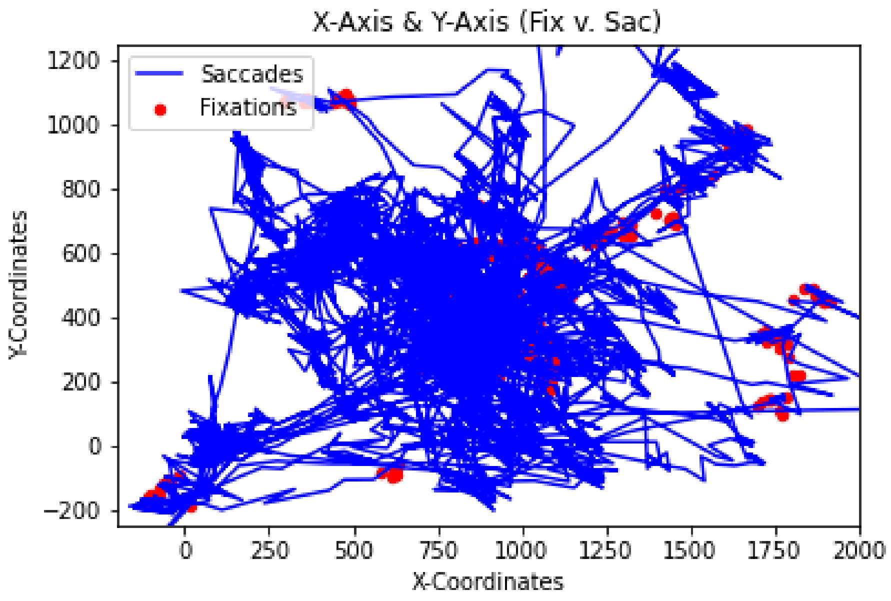


Figure 4.7: Comparing the I-AVT fixation points to the saccade paths for the I-AVT algorithm.

as a core attribute for classifying fixations and saccades, the resulting graph will contain relevant results. Since a duration threshold is explicitly set for the algorithm, there will be a clear separation between fixation and saccade results, similar to velocity graph for I-VT in 4.4. One can see from these results that the duration threshold is clearly set at 0.125 seconds, therefore any groups (or individual points) falling under that threshold are classified as blue and vice versa. The spike on the left side indicates the several individual points that were classified as saccades, and therefore all share the same duration time that in in-line with the eye-tracker capture frequency.

The resulting histograms showing the travelled distances for the fixation and saccade features respectively are found in figure 4.14. The left figure side shows the saccades overlap in the foreground, while the right figure shows fixations overlap in the foreground. When combined, a similar pattern forms to what was seen in 4.13 when comparing duration. The difference here is that some overlap occurs, but each grouping still remains quite separated and distinct. It may come as a surprise that fixations generally cover more distance compared to saccades, but this is most likely because fixations tend to have many more points in a grouping, and therefore naturally cover more total distance. Most saccade classifications are either small groups that potentially cover large distances, or single point classifications that are either classed wrongly and therefore travel fast as the fixation point. For this result, the proximity threshold was set at 70 pixels, perhaps increasing this threshold will skew the results and may potentially eliminate saccadic outliers.

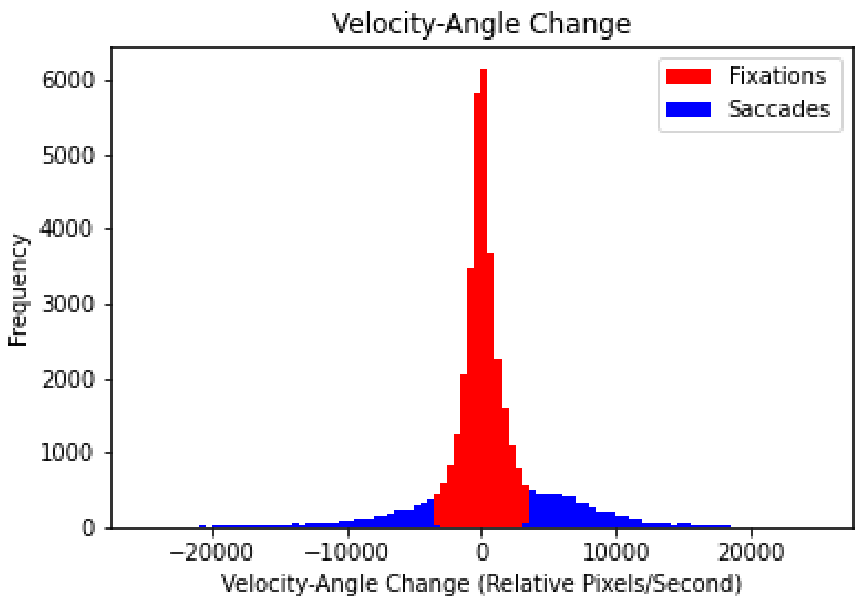


Figure 4.8: Histogram of the I-AVT angle-velocity comparisons for both the fixation and saccade points.

Figure 4.15 contains the resulting histogram, showing the angle changes for both the fixations and saccades. As the angle change classification results for the previous velocity-based algorithms were shown, it would be interesting to compare the results with a dispersion-based algorithm instead. From visual inspection, it can be seen that fixations and saccades exhibit a similar pattern of increased frequencies on each end of the angle spectrum, as well as a similar "U-shaped" graph. This could simply be due to the parameters used and procedures of the I-DT algorithm, and thus such a result can be considered as less important. It can be assumed that the results are simply inaccurate as they differ significantly from previous results, this could be due to mis-classification, and saccades are erroneously classified as fixations. One can conclude that if this mis-classification was not present, the results would closer mimic those from the other algorithms.

The results for figure 4.16 contain the scatter plot, comparing fixations and saccades to the distance travelled, with their respective durations. Since both the figures from 4.13 and 4.14 show some close relation, comparing the two key parameters involved in I-DT seems relevant. Like with the previous scatter from I-VT, this data also suffers from lack of quality. This data also shows other interesting information, where the slope of a point from the zero-axis (x and y values are both 0) is actually the velocity for that given point. In this sense, it shows that many of the classified points have varying, overlapping velocities and therefore have trivial influence on this I-DT classification. This plot is relevant strictly for I-DT due to the grouped nature of the classified features, like with 4.12.

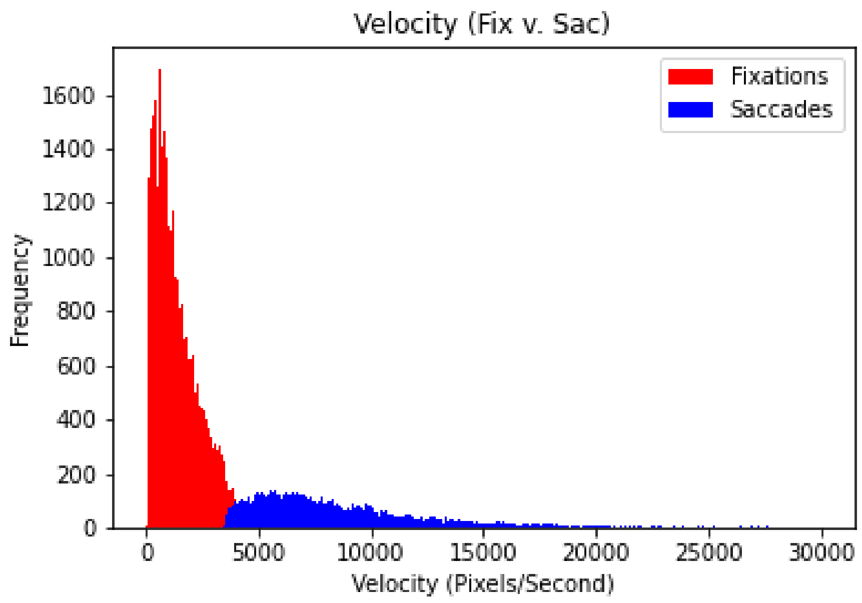


Figure 4.9: Histogram of the I-AVT velocity comparisons for both the fixation and saccade points.

Again, a clear split can be seen in the fixations and saccades where the majority of saccades are low duration and low distance, while fixations are higher duration, higher distance. Through close inspection, a couple outliers exist (saccades above the duration threshold, fixations below the duration threshold). Saccades above the duration threshold is fairly normal, as they simply did not fulfill the proximity requirement, but the series of points still lasted as long as a typical fixation would. Fixations below the duration threshold are likely anomalies when accumulating the duration of a fixation group.

4.5 Collective Algorithm Results

For all the evaluation, the eye-tracker algorithm will be considered as the ground truth for the data and discussion in these next sections.

The confusion matrices for each of the implemented algorithms are seen in figure 4.17. These confusion matrices contain the resulting numbers that are used to define the accuracy, precision, recall and F1-score calculations seen later in figure 4.18.

Accuracy is defined as the $\frac{TruePositives+TrueNegatives}{Total}$.

Precision is defined as the $\frac{TruePositives}{TruePositives+FalsePositives}$.

Recall is defined as the $\frac{TruePositives}{TruePositives+FalseNegatives}$.

Finally, F1-Score is defined as the $(\frac{1}{2}(\frac{1}{Recall}+\frac{1}{Precision}))^{-1}$

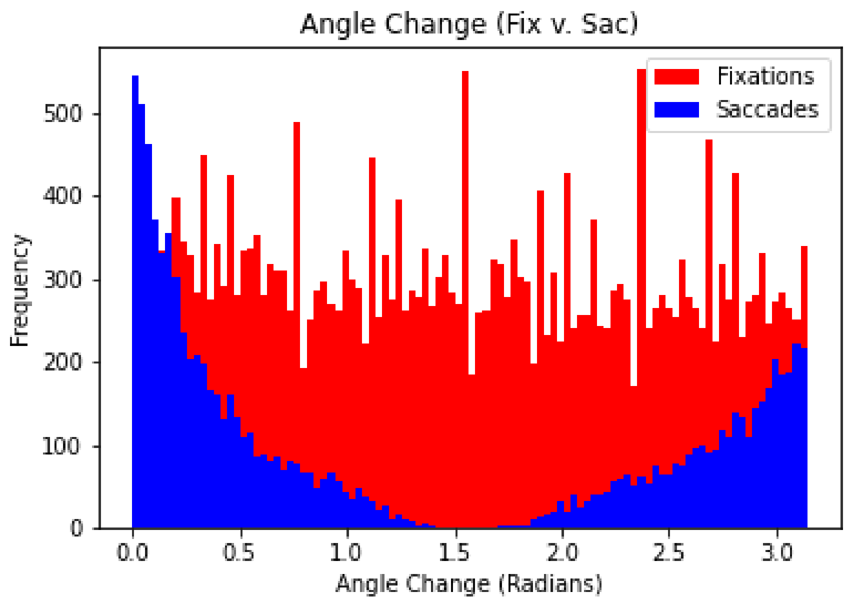


Figure 4.10: Histogram of the I-AVT angle change classifications for both the fixation and saccade points.

At first glance, the obvious common occurrence are the similar patterns in the confusion matrices for the algorithms. The most dense element in each are true positive fixations, while the weakest element are false negative outliers, being classified as saccades. One must take into account the initial uneven distribution of the points as well. Since there are more fixations than saccades, and more saccades than outliers, fixations distributions will naturally appear more dense compared to saccades. Regardless, analysing the fixations from the eye-tracker, all of the algorithms performed relatively well, with classification scores of 87.08%, 86.38%, and 87.27% for the I-VT, I-AVT, and I-DT algorithms respectively. Additionally, 'correct' saccade classification comparatively much worse, with classifications scores of 56.00%, 47.34%, and 54.27% for the I-VT, I-AVT, and I-DT algorithms respectively. Outlier results were inevitably very poor (approximately 0%) in comparison to the implemented algorithms' classification, with the I-VT only correctly classifying one instance of an outlier result. This is due to the implementations eliminating outlier classifications almost entirely.

The results table for the different evaluated parameters for all the implemented algorithms are seen in 4.18. From first glance, the results for each algorithm are quite different from each other in certain categories. Looking at the hamming distance, the I-DT performed the best with a distance of 13319 (or 66.8% of points classified correctly), while the I-AVT performed poorest with a distance of 14192 (or 64.67% of correctly classified points). This is reflected directly in the accuracy scores of the three, where the order of performance is the I-DT, then the I-VT and finally the I-AVT. The reduced accuracy scores from those presented above are certainly due to the

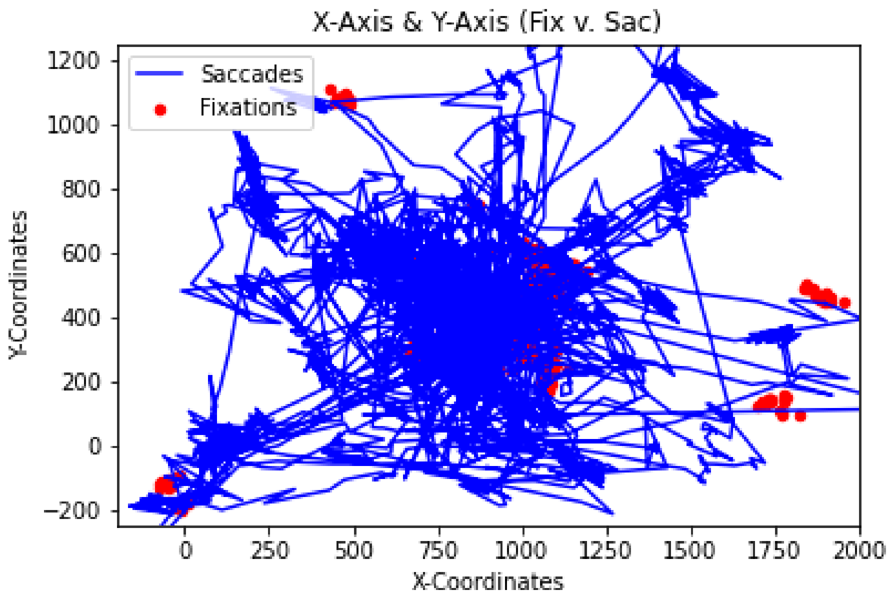


Figure 4.11: Comparing the fixation points to the saccade paths for the I-DT algorithm.

non-classification of outliers. Due to uneven distribution, the non-accuracy of outliers does not inflict a great penalty of these results. The same can be said for the above saccade classification scores, but to a lesser extent.

Next, the precision results are significantly poorer across the board, with the I-VT having the highest precision at 48%, while the I-AVT and I-DT have significantly poorer results at 39% and 40% respectively. This most likely due to the poor false positive rates of saccade classification for all of the algorithms, even though the actual number of false positive fixations are higher, the relative balance of the classification means that it has less of an impact.

Thirdly, recall performance is also rather poor across the algorithms with both the I-VT and I-DT scoring 47%, while the I-AVT scored a 45%. It can be assumed that the false negative results across all of the algorithms concerning outlier classification impacted each result equally (as they all scored the same for outlier recall). Therefore, the remaining classifications only affected the results marginally, but all were inevitably going to produce a poor overall recall result.

Finally, similar to the recall results, the F1-scores proved unimpressive across the algorithms. No algorithm proved to be significantly better than the rest in this comparison. Regardless, the I-DT scored the highest at 44%, with the I-VT and I-AVT scoring lower, at 43% and 42% respectively.

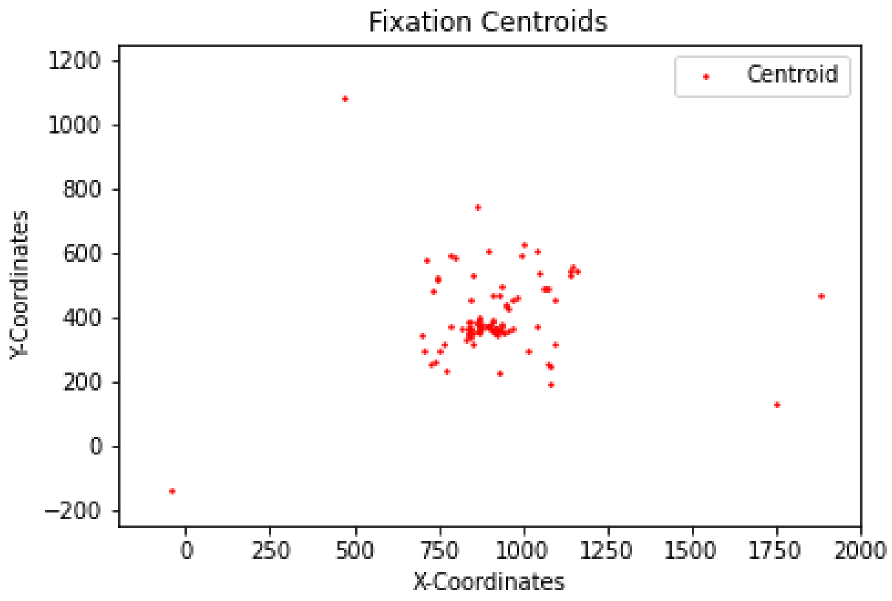


Figure 4.12: Map of the fixation centroids (for each classified fixation feature) for the I-DT algorithm.

4.6 Algorithms Evaluation & Discussion

The reason for choosing the eye-tracker algorithm as the ground truth is because we technically do not know better. As far as this thesis is concerned, the eye-tracker results are the 'correct' results, and the algorithm performance evaluation is based on how accurate the results are to the baseline. Of course, other eye-trackers could be used for this, but then they would also be the basis of evaluation. The biggest problem of course is with the amount of points classified as outliers in this instance. This makes evaluation much more difficult since the algorithms implemented were set up to exclude such results. The only exception to this rule is with the velocity threshold, where negative velocities were considered outliers as negative velocity is not possible.

A first possible measure of how good an algorithm might be, is by comparing the gaze patterns from 3.1 to 4.2, 4.7, and 4.11. This is considered a good common point of reference since all of these results share identical gaze patters, where the difference is in the actual classification of the points. Simply by looking at these graphs, the I-VT algorithm seems to have the most similar classification, while the I-DT seems to have the most dissimilar classifications as there is a clear lack of identified fixations, mainly towards the top right of the visual space. This of course is just an evaluation of a sample of the whole gaze trajectory. While it gives some idea for which algorithm is best, it is not conclusive.

As seen earlier in the matrices, the patterns exhibited are similar. This consistency is both a positive and negative result. The positive is that it

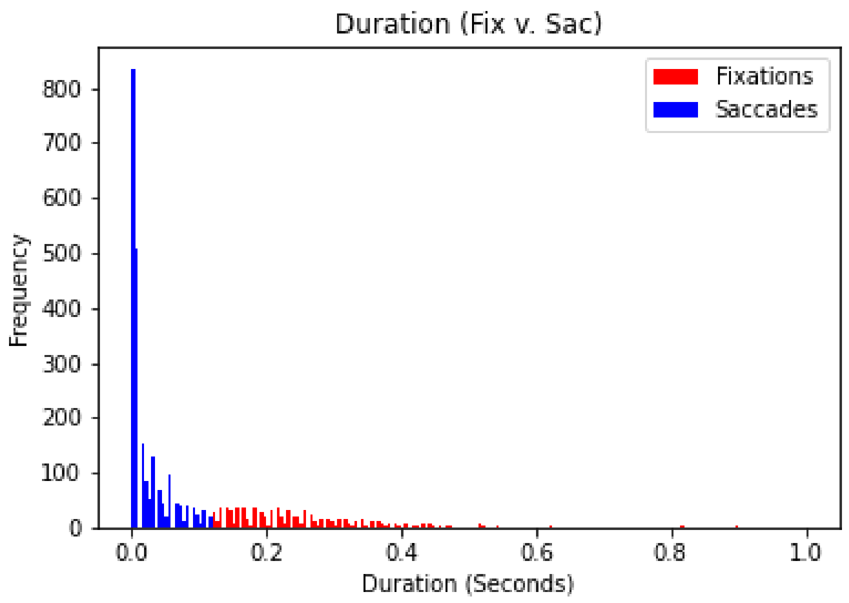


Figure 4.13: Histogram of the durations for both the fixation and saccade features.

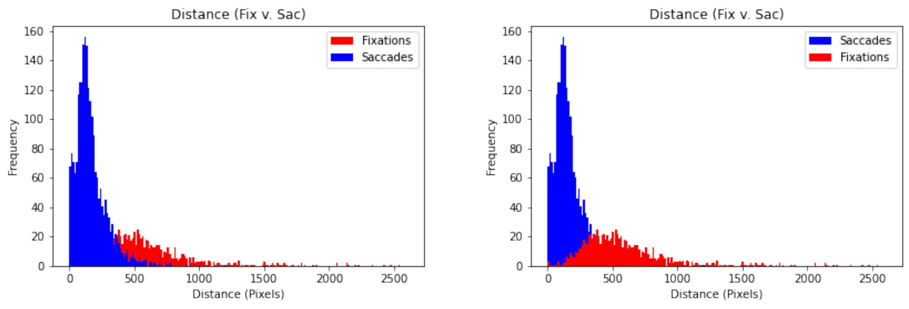


Figure 4.14: Histograms of the total travelled distances for both the fixation and saccade features.

shows that all of the algorithms have consistent patterns, and there are no serious anomalies in terms of classifying 'incorrectly'. The drawback is that if any one of these results are relatively poor, the other algorithms will also end up poor. If one ended up having dramatically different results, there is a chance that those results may have ended up being closer to the truth.

Some correlation observation attempts were made on some of the parameters, specifically to see how much these crucial parameters affected each other. These observations were done in figures 4.5 and 4.16. It was determined that these observations were not relevant for the specific algorithm evaluation, but may still be worth observing in a deeper investigation. It was determined more important to look at the classification in these plots, seeing where the classifications are among the scatters, and what can be learned from them. As noted in the respective results sections, it was worth

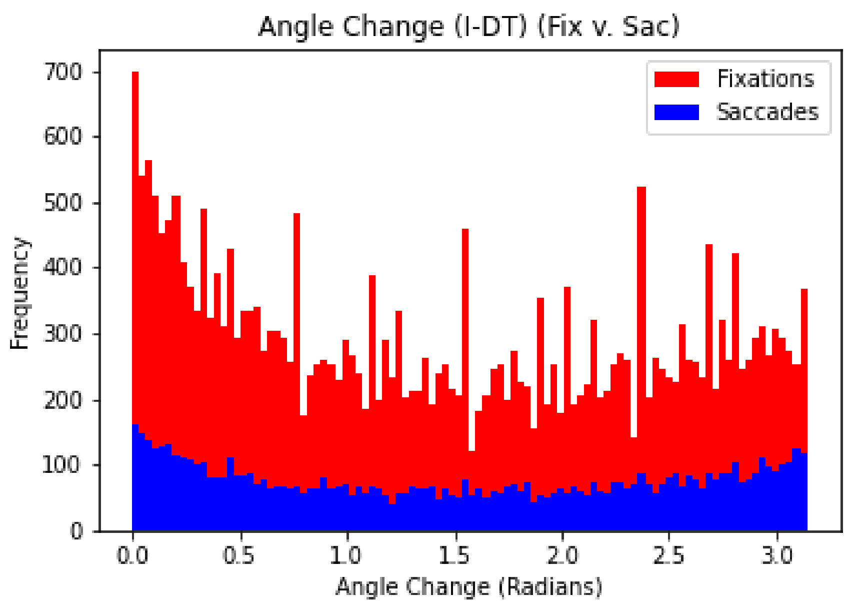


Figure 4.15: Histogram of the I-DT angle change classifications for both the fixation and saccade points.

noting how these diagrams show a lack of quality data, being that many points are scattered, not due to naturally occurring outliers, but rather because data is missing in between the points plotted. These correlation observations do not exist for the I-AVT plot as it was determined to not be a priority. However, considering how the angle parameter affected the results compared to a standard I-VT, a further investigation into this relationship is worth looking into. The last point of note for correlation is the confusion matrix results from fig 4.18.

A possible hypothesis to be made is that the best performing algorithm should score the best results in each respective category, but this is not the case here. There is no correlation between these results (for example, if one algorithm has a much higher precision score than the others, it will not necessarily score the best from an accuracy perspective.) .Therefore, a singular algorithm cannot be determined as the best one relative to the eye-tracker. What can be observed instead, is that certain algorithms perform better than others in certain areas. Ever further than that, a best possible algorithm would be to combine aspects of each algorithm, by determine where their strengths are, and integrating them into one comprehensive algorithm. A problem with this scenario, is that this ideal algorithm may perform worse to a typical algorithm with some other eye-tracker data or equipment. This is specifically why certain algorithms discussed in the background chapter (N&H for example) work best in their controlled environments, but not necessarily from a general perspective. So perhaps an adaptive algorithm is the best choice.

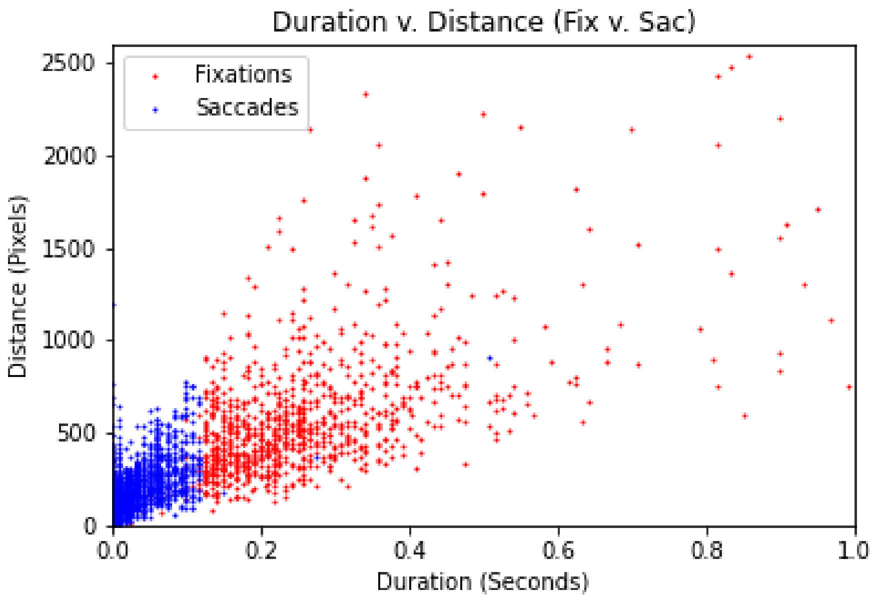


Figure 4.16: Scatter plot comparing both the fixation and saccade points in relation to total distance travelled and total duration simultaneously.

There were some expectations to be had with the K-Ratio in the hopes that it would improve the existing results, but unfortunately, it did not affect them much. When it came to implementing it, doing so for the I-VT and I-AVT were easy as they both only have one parameter, therefore the K-Ratio is only working on one dimension. For the I-DT however, implementing it was possible, but it seems having two parameters that may act against each other for ideal results causes problems when finding the K-Ratio. It was clear that the lowest K-Ratio values applied to the I-DT resulted in extreme results where the ratio between fixations and saccades was too high (e.g. 20 saccades to 1 fixation).

Regardless, the K-Ratio results were ultimately not implemented in for the results above. This is simply due to how similar the predicted threshold from visual observation were from the K-Ratio (E.G the lowest K-Ratio for I-VT was 4368 pixels per second, similar to the estimated 4200 pixels per second). For context, the visually observed thresholds applied originally were based on getting as close to the eye-tracker's fixation to saccade ratio (approximately 4:1) as possible. The results from the confusion matrix, as well as the classification report, and hamming distance, were not any better than the results prior to implementing the K-Ratio values. One can infer that the K-Ratio is better suited for algorithms where a clear "ideal" threshold is unclear, like in the I-DT, despite the results not proving useful. The I-DT thresholds are based on groups of fixations and saccades, while the thresholds for the other two are based on individual points, this may be why the K-Ratio was not working ideally.

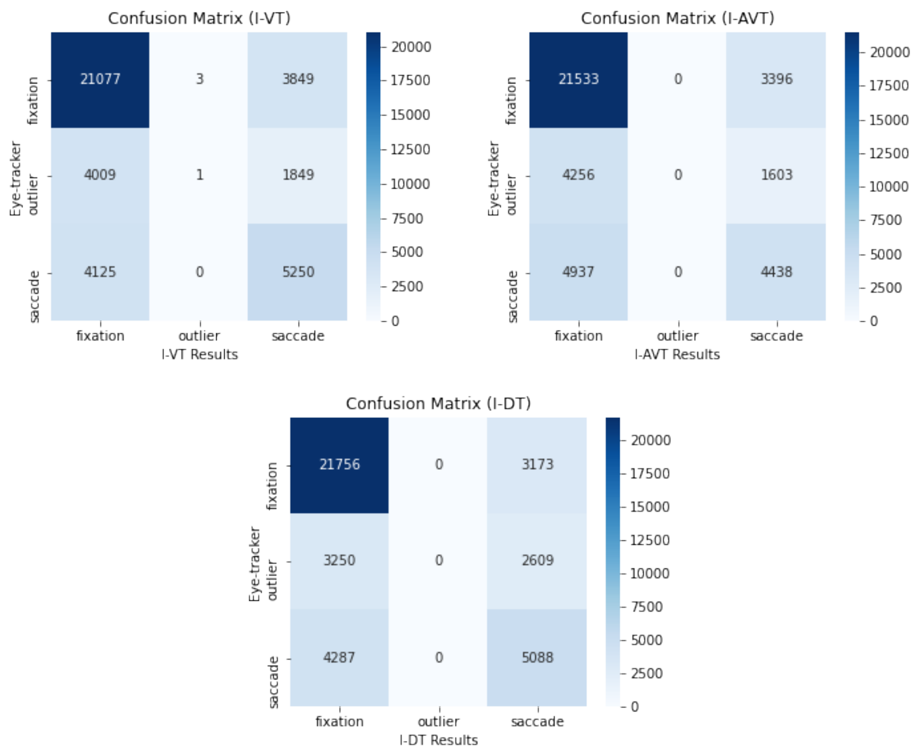
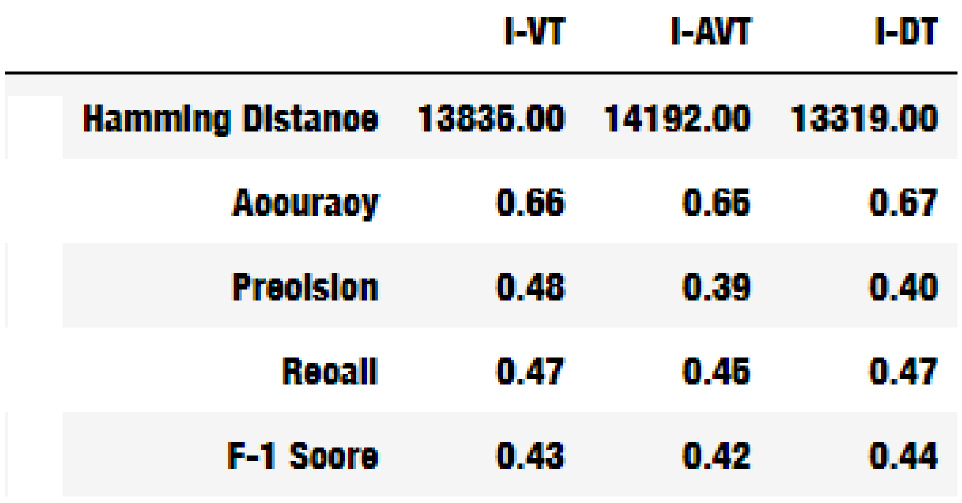


Figure 4.17: Confusion matrices for the I-VT, I-AVT, and I-DT algorithms respectively.



	I-VT	I-AVT	I-DT
Hamming Distance	13835.00	14192.00	13319.00
Accuracy	0.66	0.66	0.67
Precision	0.48	0.39	0.40
Recall	0.47	0.46	0.47
F-1 Score	0.43	0.42	0.44

Figure 4.18: A table comparing the matrix results of the algorithms.

Chapter 5

Conclusions & Future Work

To summarise the work done in the thesis, several different algorithmic approaches were used with the goal of determining which of them are the best for classifying fixations and saccades, based on the original eye-tracker. Implementation difficulty proved to not be relevant in terms of how the algorithms performed, as something very simple like the I-VT did not perform any worse compared to the more complex I-AVT and I-DT algorithms. This of course can change when moving to far more advanced algorithms like I-HMM or I-MST, where the implementation is very robust and complex. Naturally, one expects these algorithms to produce relatively great results over what has already been implemented. On the whole, the I-VT algorithm turned out to be the best algorithm. Just from comparing the accuracy and recall results, it may seem inconclusive, but its significantly higher precision score makes it the best choice in this case. Perhaps on a different experiment, the results will be more convincing, but not in this case. Since only two features were distinguished from the eye-tracking results, this may have affected how the tests are concluded. There is a chance that much of the data are blinks or noise, thus classifying them as something discrete is incorrect interpretation, hence why the eye-tracker algorithm classified them as outliers. Therefore, this automatically makes the algorithms implemented with the discrete fixation and saccade classifications lose relative performance over the baseline.

Perhaps better technology will make future eye-tracking classification better, and evaluating existing, proven algorithms will prove more useful when compared with an equally strong classifier within the eye-tracking devices themselves. A better eye-tracker, paired with a potentially improved algorithm, to be used in the future will hopefully classify eye-gaze points more accurately and with more discretion, to avoid having as many outliers points as there were in this instance. This will obviously help with external algorithmic evaluation, similar to what is being done here. In this case however, combining the methods is perhaps the best way to approach the problem. Perhaps if these tests were done on more than one data set simultaneously, more concrete results may have emerged from these findings. Another suggestion for the future is a robust way to determine the results of algorithms. In this thesis, several different visualizations and results were projected to determine the performance of the algorithms. Perhaps a standard approach for classification results are not ideal for eye-tracking classification specifically, and better methods need to be developed. This can also be extended to the proposed ideal solution for finding the best possible algorithm, via the K-Ratio. The results with its usage can be considered disappointing as they did not yield any serious improvements in the results. With this in mind, perhaps a better implementation of the K-Ratio is necessary. This can be done using more complicated and varying data sets, the use of other algorithms not implemented in the thesis, or simply by tweaking the K-Ratio algorithm itself.

Bibliography

[1] Hamed Rezazadegan Tavakoli, Esa Rahtu and Janne Heikkilä. 'Stochastic bottom–up fixation prediction and saccade generation'. In: *Image and Vision Computing* 31.9 (2013), pp. 686–693.

[2] Dario D Salvucci and Joseph H Goldberg. 'Identifying fixations and saccades in eye-tracking protocols'. In: *Proceedings of the 2000 symposium on Eye tracking research & applications*. 2000, pp. 71–78.

[3] Alexandra I Korda et al. 'Automatic identification of eye movements using the largest lyapunov exponent'. In: *Biomedical Signal Processing and Control* 41 (2018), pp. 10–20.

[4] Linnéa Larsson, Marcus Nyström and Martin Stridh. 'Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit'. In: *IEEE Transactions on biomedical engineering* 60.9 (2013), pp. 2484–2493.

[5] Teresa Busjahn et al. 'Eye tracking in computing education'. In: *Proceedings of the tenth annual conference on International computing education research*. 2014, pp. 3–10.

[6] David A Slykhuis, Eric N Wiebe and Len A Annetta. 'Eye-tracking students' attention to PowerPoint photographs in a science education setting'. In: *Journal of Science Education and Technology* 14.5 (2005), pp. 509–520.

[7] Pierre Chandon et al. 'Measuring the value of point-of-purchase marketing with commercial eye-tracking data'. In: *INSEAD Business School Research Paper* 2007/22 (2006).

[8] Michel Wedel and Rik Pieters. *Eye tracking for visual marketing*. Now Publishers Inc, 2008.

[9] Anneli Olsen. 'The Tobii I-VT fixation filter'. In: *Tobii Technology* 21 (2012).

[10] Marcus Nyström and Kenneth Holmqvist. 'An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data'. In: *Behavior research methods* 42.1 (2010), pp. 188–204.

[11] Raimondas Zemblys et al. 'Using machine learning to detect events in eye-tracking data'. In: *Behavior research methods* 50.1 (2018), pp. 160–181.

[12] Oleg V Komogortsev and Alex Karpov. 'Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades'. In: *Behavior research methods* 45.1 (2013), pp. 203–215.

[13] Soussan Djamasbi. 'Eye tracking and web experience'. In: *AIS Transactions on Human-Computer Interaction* 6.2 (2014), pp. 37–54.

[14] Pontus Olsson. *Real-time and offline filters for eye tracking*. 2007.

[15] Daniel C Richardson and Michael J Spivey. 'Eye tracking: Characteristics and methods'. In: *Encyclopedia of biomaterials and biomedical engineering* 3 (2004), pp. 1028–1042.

[16] Kyle Krafka et al. 'Eye tracking for everyone'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2176–2184.

[17] Ignace TC Hooge et al. 'Is human classification by experienced untrained observers a gold standard in fixation detection?' In: *Behavior Research Methods* 50.5 (2018), pp. 1864–1881.

[18] Dario D Salvucci. 'An interactive model-based environment for eye-movement protocol analysis and visualization'. In: *Proceedings of the 2000 symposium on eye tracking research & applications*. 2000, pp. 57–63.

[19] Richard Andersson et al. 'One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms'. In: *Behavior research methods* 49.2 (2017), pp. 616–637.

[20] Richard Schweitzer and Martin Rolfs. 'An adaptive algorithm for fast and reliable online saccade detection'. In: *Behavior research methods* 52.3 (2020), pp. 1122–1139.

[21] Krzysztof Krejtz et al. 'Entropy-based statistical analysis of eye movement transitions'. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. 2014, pp. 159–166.

[22] Pieter Blignaut. 'Fixation identification: The optimum threshold for a dispersion algorithm'. In: *Attention, Perception, & Psychophysics* 71.4 (2009), pp. 881–895.

[23] Pedro Lencastre et al. 'EyeT4Empathy: Database of foraging for visual information, visual writing and empathy assessment'. In: *Behavior research methods* (2021).

[24] Samip Bhurtel, Pedro G. Lind and Gustavo B. Moreno e Mello. 'For a New Protocol to Promote Empathy Towards Users of Communication Technologies'. In: *HCI International 2021 - Late Breaking Posters*. Ed. by Constantine Stephanidis, Margherita Antona and Stavroula Ntoa. Cham: Springer International Publishing, 2021, pp. 3–10. ISBN: 978-3-030-90176-9.

[25] G David Forney. 'The viterbi algorithm'. In: *Proceedings of the IEEE* 61.3 (1973), pp. 268–278.

[26] Leonard E Baum et al. 'A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains'. In: *The annals of mathematical statistics* 41.1 (1970), pp. 164–171.

[27] Oleg V Komogortsev et al. 'Standardization of automated analyses of oculomotor fixation and saccadic behaviors'. In: *IEEE Transactions on biomedical engineering* 57.11 (2010), pp. 2635–2645.

[28] Ralf Van der Lans, Michel Wedel and Rik Pieters. 'Defining eye-fixation sequences across individuals and tasks: the Binocular-Individual Threshold (BIT) algorithm'. In: *Behavior research methods* 43.1 (2011), pp. 239–257.

[29] Giacomo Veneri et al. 'Eye fixations identification based on statistical analysis-case study'. In: *2010 2nd International Workshop on Cognitive Information Processing*. IEEE. 2010, pp. 446–451.

[30] PM Camerini, G Galbiati and F Maffioli. 'Algorithms for finding optimum trees: Description, use and evaluation'. In: *Annals of Operations Research* 13.1 (1988), pp. 263–397.

[31] Josef Krivan. *GitHub Project Page*. https://github.com/Josef733/eyetracker Josef733/eyetracker. [Online]. 2022.