Erasmus University Rotterdam

Master Thesis Econometrics

Erasmus School of Economics,
MSc Econometrics

Extending eye fixation classification algorithms through variety within tasks

Author:

Jonathan Schoenmaker (596360)

Supervisor:

Dr. A. Alfons

Second Assessor:

Dr. A. Martinovici

July 9, 2022

Abstract

Being able to accurately assess the location of eye fixations of a subject is essential for various fields, from medical applications to marketing. This paper compares multiple extensions of a prominent eye fixation detection algorithm. The extensions are expanded versions of the original through their ability to determine whether a subject is gazing at an image or a piece of text within the context of a task. Results show there is a difference in eye-movements when switching between images and text within a task for all extensions. These differences, however, are many times smaller than the differences that are caused by variations between individuals. Moreover, results show that the extensions perform differently than the original algorithm, without a large difference in fixation classification. There is no notable difference between the extensions and the original in terms of fixation length or classification of the fixations with the longest duration, with the exception of the longest fixation merging with the preceding fixation in all extensions. Restricted versions of the extensions created limited a fixation to one region, image or text. These restricted versions of the extensions scored worse on the percentage of unrealistic fixations classified, compared to the original. The unrestricted versions, where fixations are allowed to continue across different regions, scored equal compared to the original.

Contents

1 Introduction

Eye-tracking applications occupy a wide field of studies. Being able to accurately assess the location of a subject's fixation is critical for marketing studies, psychological experiments and vision research. If a company wishes to evaluate their new advertisement poster, they use eye-tracking data to determine which parts of their poster are noticed first or looked at most frequently. Medical experiments observe eye movements that do not cover the full distance from point of interest A to B (hypometric saccades) as a possible indicator of Parkinson's disease, among others. (Termsarasab, Thammongkolchai, Rucker, & Frucht, 2015).

The coordinates of a subjects viewing direction on the screen is known as the Point of Regard (POR). In order to determine the properties of a subjects eye movements when gazing at a POR, eye tracking algorithms aim to differentiate eye fixations from outliers such as saccades, blinks and other anomalies caused by errors in measurement equipment. Currently, two main methods of differentiating fixations from outliers are used: dispersion-based methods and velocity-based methods (Holmqvist et al., 2011). Dispersion based methods differentiate between fixations and outliers by examining the distance the eye travels between measurements, this distance is compared to a threshold. The new measurement is classified as either a possible fixation or an outlier. Velocity-based methods follow a similar approach, but instead of examining the distance between measurements, the algorithm examines velocity and acceleration of the eye, and classifies measurements as a possible fixation or outlier based on threshold values. Due to their increased transparency and accuracy velocity-based methods are often seen as the preferred method (Holmqvist et al., 2011).

Eyes have different properties of fixations and saccades depending on the person, the eye, or the task that the subject is asked to perform. For example, a subject searching for a particular object in an image has larger saccade lengths on average than a person reading a text (Van der Lans, Wedel, & Pieters, 2011). Algorithms need methods to adjust for these differences. Current velocity-based methods try to differentiate between individuals, eyes and tasks. The algorithm by Engbert and Mergenthaler achieves this but does not take into account the relationship between movements of the separate eyes (Engbert & Mergenthaler, 2006). The algorithm of Nyström and Holmqvist only makes use of monocular data, meaning they only use data from one eye or the average of both eyes, rather than using data from both eyes (binocular) separately (Nyström & Holmqvist, 2010). One of the velocity-based algorithms that outperforms or competes with other algorithms and solves the issues mentioned above when detecting fixations in image and video data is the Binocular-Individual Threshold (BIT) algorithm (Nyström & Holmqvist, 2010). This algorithm makes use of robust minimum covariance estimators to differentiate fixations from outliers (Van der Lans et al., 2011).

BIT estimates the threshold velocity in both the x and y directions. These thresholds are estimated for both eyes. For each participant and task the algorithm is run individually, allowing for different estimates to arise specific to participants and tasks (Van der Lans et al., 2011). More details about the workings of the BIT algorithm are explained in chapter 3 of this paper. The BIT algorithm is able to calculate these thresholds specific to variables that change infrequently. A participant is not substituted halfway through an experiment only to come back later, and a task is not changed halfway through an experiment. The algorithm is unable to calculate thresholds specific to variables that do change frequently. There is room for improvement within these algorithms to account for frequently changing variety within experiments. These variations can be split into two groups: observed and unobserved variations. An example of unobserved variations is an individuals attention state. When a subject is scanning a complex scene they switch between a local and global attention state, both of which have differing properties of fixations and saccades (Liechty, Pieters, & Wedel, 2003). The reason this is an unobserved variation is that directly measuring a subject's attention state is impossible. In other words, the attention state is not an input variable to the algorithm. An example of an observed variation that is not accounted for in the BIT algorithm is information density. Within a task, information might vary largely per position, some areas of the screen might not hold any information and other areas might contain a high amount of information. Since eye-tracking data provides the coordinates of the POR, it is possible to define high and low information areas depending on the subject's POR. This can be used as an input and then differentiated when determining fixations. If algorithms are specific to more variability, the performance of these algorithms in identifying fixations from saccades might improve. This research focuses on extending the BIT algorithm to include variability within tasks, with a specific focus on including observed variations for comparing areas with images versus areas with text.

An important caveat to the difference between these observed variations and variations of already existing elements, is that the observed variations are able to switch back and forth during an experiment. Making a task-specific threshold is easier to accomplish, since each task can be easily separated in different data sets for calculation. However, when there is a constant switching within the experiment, this separation is not easily achieved without creating long breaks in the data. Furthermore, there are uncertainty periods within the data, where the behaviour might not reflect either of the states it is in, that need to be taken into account. With this in mind, the goal of this paper is to extend the BIT algorithm to include variety within tasks by adjusting for switching from images to text and back. The goal is to have the researcher define the region of the image within the experiment without any further

need to split the data. To evaluate the performance of the extended BIT algorithms the following research questions are formulated: Is there a significant difference in eye-movement when switching between images and text within a task? And secondly: What are the main differences in fixation classification between the original and the extended BIT algorithms?

The results show there is a difference in eye-movements when switching between images and text within a task for all extensions. These differences are magnitudes smaller than the differences that are caused by variations between individuals. Results also show that the extensions generate some different fixations than the original algorithm, with a large group of fixations being identically classified across both the extended algorithms and the original. There is no notable difference between the extensions and the original in terms of fixation length. The only metric for performance used in this research is the percentage of unrealistic fixations classified. All extensions performed similarly compared to the original according to this metric. In order to assess the quality of the performance of the extensions the algorithms must be applied to manually checked data-sets. In these data-sets all stamps are classified as being part of a fixation or not. If these algorithms are applied to these data-sets, a percentage score of correct fixations is a good indicator of quality of performance.

The structure of the remainder of this thesis is as follows: Chapter 2 provides a detailed explanation of the data used to evaluate the extended algorithm. Chapter 3 dives into the methodology of both the original BIT algorithm as well as the extensions. Chapter 4 discusses the results of all the different extensions used compared to the original algorithm and finally chapter 5 draws a conclusion from the results analysed in chapter 4 and attempts to answer the research questions.

2 Data

The data used for this research is provided by a 2016 consumer choice study where a subject is prompted with a task to choose a brand from four different options in four different product categories. An example of a slide for one of these product categories is shown below.

	BRAND 1  Natur	BRAND 2  Oral-B	BRAND 3  Preserve	BRAND 4  Jordan
Handle	Recycled wood	Plastic	Recycled plastic	Plastic
Bristles	Natural hair	Nylon	Nylon	Nylon
Whitening	No	Yes	No	No
Rubber grip	No	Yes	Yes	Yes
Tongue cleaner	No	No	No	Yes

Figure 1: An example of a slide used in the study for one particular product category.

Before a subject is prompted with one of these slides, they are directed to choose an item based on a particular quality, they are informed that all products have similar price/quality ratios and fit their budget. In the example above the subject might be assigned the task to select the most environmentally friendly product, this task differs between subjects. In the example above subjects see the slide for a fixed time of 15 seconds before moving on, this is the time-pressure variant of the slide. Another variation exists with a showtime of 30 seconds. Finally, the order that the brands appear in have two different variations. The first variation starts with brand 1 on the left side and counts up to brand 4. The other variations starts with brand 4 on the left and counts down to brand 1. There is a total of 5 slides to be analysed with 8 different experimental conditions. Each of these unique combinations is separated into its own data-set, resulting in 40 different combinations with the relevant variables as shown in the table below. All results in this research are generated from one of these 40 combinations, namely the data on slide 5 which shows information on toothbrushes. This combination does not include time pressure and shows the brands in the order from 1-4 as shown in the example

above. This combination includes 131,630 observations from 72 individual participants. The difference between two subsequent observations by the same participant is 17ms. Only one of these combinations was selected since the focus of this research is not on analysis of the actual consumer study, but rather differences between the BIT algorithm and the extensions. To best evaluate this it is ideal for all data to be generated from the same experimental condition.

Table 1: Relevant variables in the data sets

Variable Name	Description
Participant ID	A numerical patient identifier
Media Name	The filename of the relevant slide the subject is watching
MediaXPosition	The x-coordinate of the top-left of the slide
MediaYPosition	The y-coordinate of the top-left of the slide
RecordingTimeStamp	Time since recording started in ms
GazePointLeftX	x position of the gaze of the left eye
GazePointLeftY	y position of the gaze of the left eye
GazePointRightX	x position of the gaze of the right eye
GazePointLeftY	y position of the gaze of the right eye
DistanceLeft	The distance of the left eye to the screen
DistanceRight	The distance of the right eye to the screen
Note:	DistanceLeft and DistanceRight contain a mistake and are equal for all observations

The BIT algorithm takes 7 variables as input when utilising binocular data, and 5 variables when using monocular data. Firstly, the BIT algorithm separates participants based on their *Participant ID*. It uses the differences in *RecordingTimeStamp* to determine the duration of a single observation. To analyse behaviour of the eyes the algorithm uses the x and y coordinates for both eyes with respect to the top left corner of the media shown, along with the distance from the eyes to the screen, which is a single variable and does not differ between eyes. When utilising monocular data there is only a single coordinate for x and y position. This can either be the coordinate of a single eye, or the averaged coordinates of both eyes.

In order to adjust the data to fit the BIT algorithm, the relative gaze position from the top-left of the slide is calculated by subtracting the *GazePoint* variables from their respective *MediaPosition* variables, resulting in *EyePos(Left/Right)(X/Y)Relative* as new variables. Furthermore, since *DistanceLeft* and *DistanceRight* are equal for all observations, they are combined into a new variable *Distance*. This results in the following variables being used in the original BIT algorithm: *Participant ID, RecordingTimeStamp, EyePosLeftXRelative, EyePosLeftYRelative, EyePosRightXRelative, EyePosRightYRelative, Distance*.

For the purpose of this research, an additional column is added to specify whether the position a subject is gazing is located a image/text area (*Image*). Each of these areas are defined by the surrounding black box. In the previous example of a slide, this would mean that if a participant is gazing to the top row of the slide, they get a value of $Image = 1$, where a gaze to the other five rows results in the value $Image = 0$. Large parts of this research focus on

eye-averaged data, where the variables used are the average x and y positions of the eyes.

3 Methodology

An important aspect of the BIT algorithm is that it estimates the velocity-threshold using robust statistics instead of setting a pre-determined threshold to differentiate fixations from outliers. The eye movements are logged in the algorithm through $\boldsymbol{z}_t = (x_{t,left}, y_{t,left}, x_{t,right}, y_{t,right})'$ for $t = 1, ..., T$. Since the algorithm is velocity-based, the first difference $\Delta(\boldsymbol{z}_t) = (\boldsymbol{z}_t - \boldsymbol{z}_{t-1})$ is used to determine the velocity threshold. BIT assumes that eye velocities (when fixating) are distributed through a multivariate normal distribution, with a separate mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ per participant, eye and task. These velocities are called the within-fixation velocities (Van der Lans et al., 2011). The velocity thresholds determined by the algorithm form a tolerance ellipse, which determines whether observations are possibly part of fixations. In the section below, the BIT algorithm is described in depth. This is followed by an explanation of the extensions using image labelling, uncertainty periods, and confidence weights.

3.1 The BIT algorithm

The BIT algorithm uses the observations $\boldsymbol{z}_t$ as described above along with subject IDs and a timestamp corresponding to each observation. The algorithm is run separately for each unique participant ID and proceeds to check the validity of all these observations. If any observation $\boldsymbol{z}_t$ does not fall within the boundaries $x_t, y_t \geq 0$, $x_t \leq x_{max}$ and $y_t \leq y_{max}$, where y_{max} and x_{max} are set by the edges of the slide that is being viewed, this observation is given a label that indicates it is not a valid observation. The valid observations are then run through the fast minimum-covariance determinant (FAST-MCD) algorithm to approximate the value of the MCD estimator (Rousseeuw & Driessen, 1999).

3.1.1 MCD

The aim of the MCD estimator is to find robust estimates of a multivariate location and scatter that approximate the multivariate normal distribution of the within-fixation velocities. In a non-robust situation, all valid observations are included in determining the mean and covariance of the multivariate normal distribution. This makes the estimation extremely vulnerable to outliers. In this research all non-fixations in the observations are seen as outliers from the multivariate normal distribution. A method to deal with these outliers when determining the location and scatter is necessary. We start with the idea of two different distributions. One of which is a multivariate normal that one wants to analyse, the other containing outliers generated by another (unknown) distribution. This approach is called the Tukey-Huber contamination model (Huber, 1992) and is denoted by:

$$Q = (1 - \epsilon)\Phi + \epsilon G, \tag{1}$$

where Φ is the distribution one wishes to analyse, G is the contaminating distribution and ϵ is size of the contamination in the mixture distribution. The main idea of the MCD estimator is to take a subset of h observations such that $[(T+p+1)/2] \leq h \leq T$, for which the determinant of the covariance matrix is minimal. Here p is the amount of explanatory variables. This subset size h can be interpreted as the sample analogue of the $1-\epsilon$ factor in the Tukey-Huber model. The objective function to find the appropriate subset H is shown below

$$H_{MCD} = \arg\min \det(\boldsymbol{S}_H), \tag{2}$$

with the specification $H : |H| = h$. Here $\boldsymbol{S}_H$ is the sample covariance of the subset H, with corresponding sample mean t_H. If the selected subset is H_{MCD} then the MCD estimator of location $t_H = t_{MCD}$ is Fisher consistent for the mean $\boldsymbol{\mu}$ under the normal distribution Φ. The MCD estimator for scatter $\boldsymbol{S}_H = \boldsymbol{S}_{MCD}$ however, is underestimated. This can be corrected by two correction factors. The first one is the Fisher consistency correction, which is shown below.

$$c_\alpha = \frac{\alpha}{F_{\Gamma(\frac{p}{2}+1,1)}(\frac{\chi^2_\alpha}{2})}, \tag{3}$$

where α is the relative size of the subset H compared to the full set. $F_{\Gamma(\frac{p}{2}+1,1)}$ is the cumulative distribution function of the gamma distribution $\Gamma(k,\theta)$ with $k = (p/2)+1$ and $\theta = 1$, and χ^2_α is the $\alpha-$quantile of the $\chi^2(p)$ distribution. The second correction factor is the small sample correction c_{np} which approaches 1 if the number of observations goes to infinity (Pison, Van Aelst, & Willems, 2002). With these corrections it holds that $\hat{\boldsymbol{\mu}}_{MCD} = t_{MCD}$ and $\hat{\boldsymbol{\Sigma}}_{MCD} = c_\alpha c_{np} \boldsymbol{S}_{MCD}$.

The standard tolerance ellipse is constructed through the Mahalanobis distance

$$D_M(\Delta(\boldsymbol{z}_t)) = \sqrt{(\Delta(\boldsymbol{z}_t) - \bar{\Delta}(\boldsymbol{z}_t))' \boldsymbol{S^{-1}} (\Delta(\boldsymbol{z}_t) - \bar{\Delta}(\boldsymbol{z}_t))}, \tag{4}$$

where $\boldsymbol{S}$ is the covariance matrix of the sample. Since this measure uses both the sample mean and the sample covariance matrix, one extreme outlier ($|\boldsymbol{z}_t| \longrightarrow \infty$) causes the estimators for the sample mean and sample covariance to break down. Similarly in the multivariate case, the masking effect comes into play which states that large outliers might not have large Mahalanobis distances (Rousseeuw & Driessen, 1999). This method of constructing a tolerance ellipse is therefor not robust. The MCD algorithm resolves this issue by constructing the ellipse through a robust distance

$$D_R(\Delta(\boldsymbol{z}_t)) = \sqrt{(\Delta(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_{MCD})' \hat{\boldsymbol{\Sigma}}^{-1}_{MCD} (\Delta(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_{MCD})}, \tag{5}$$

where $\hat{\boldsymbol{\mu}}_{MCD}$ is the MCD location estimate and $\hat{\boldsymbol{\Sigma}}_{MCD}$ is the MCD scatter estimate obtained through the correction of $\boldsymbol{S}_{MCD}$. These estimates are defined through:

$$\hat{\boldsymbol{\mu}}_{MCD} = \frac{\sum_{t=1}^{T} W(d_t^2)\Delta(\boldsymbol{z}_t)}{\sum_{t=1}^{T} W(d_t^2)} \tag{6}$$

and

$$\boldsymbol{S}_{MCD} = \frac{1}{n}\sum_{t=1}^{T} W(d_t^2)(\Delta(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_{MCD})(\Delta(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_{MCD})', \tag{7}$$

where $W(d_t^2) = I(d_t < \chi^2_{p,0.975})$ is a weight function based off the chi-squared distribution and

$$d_t(\Delta(\boldsymbol{z}_t)) = \sqrt{(\Delta(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_0)'\hat{\boldsymbol{\Sigma}}_0^{-1}(\Delta(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_0)}, \tag{8}$$

where $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\Sigma}}_0$ are the sample location and scatter for the subset H (Rousseeuw & Driessen, 1999). This procedure detects points that have a probability below a certain threshold to be generated from the aforementioned multivariate normal distribution. The figure below shows an example of the differences between D_R and D_M.

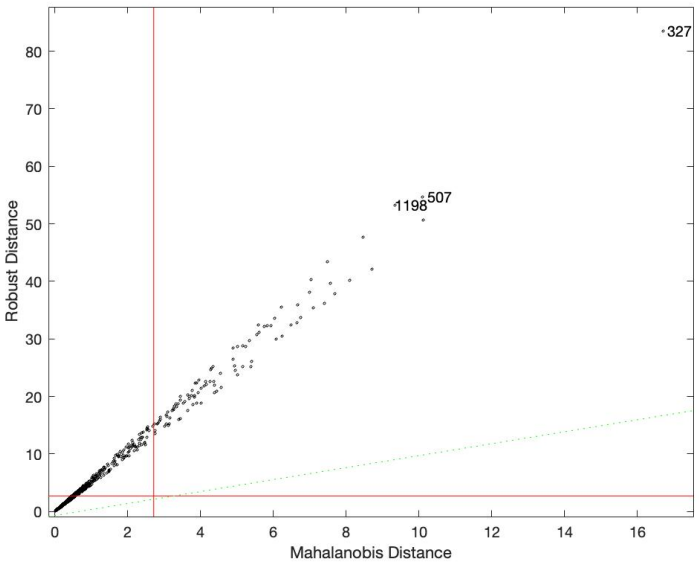


Figure 2: The Robust and Mahalanobis distances of Participant 7.

3.1.2 FAST-MCD

The FAST-MCD algorithm approximates the MCD estimator by taking a different approach in selection and sequencing of subsets h. The MCD estimator needs the computation of every possible subset h, which leaves $\binom{n}{h}$ iterations of computation, quickly resulting in an orders of magnitude slower approach than FAST-MCD. The FAST-MCD algorithm benefits from the concentration step (C-step). The idea behind this step is to first select a random subset H_1 of

observations of size h. For this subset we calculate $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\Sigma}}_1$. The next step is to calculate the robust distances for all observations

$$d_1(t) = \sqrt{(\Delta(\boldsymbol{z}_i) - \hat{\boldsymbol{\mu}}_1)' \hat{\boldsymbol{\Sigma}}_1^{-1} (\Delta(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_1)}, \text{ for } t = 1, 2, ..., T. \tag{9}$$

Based on these distances, the subset H_2 is constructed such that it selects the h observations with the smallest robust distances to $\hat{\boldsymbol{\mu}}_1$. For the new subset $\hat{\boldsymbol{\mu}}_2$ and $\hat{\boldsymbol{\Sigma}}_2$ are calculated. Because the subset H_2 always selects the smallest distances possible to $\hat{\boldsymbol{\mu}}_1$ it holds that

$$\det(\hat{\boldsymbol{\Sigma}}_2) \leq \det(\hat{\boldsymbol{\Sigma}}_1), \tag{10}$$

where equality is only reached if $\hat{\boldsymbol{\mu}}_2 = \hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\Sigma}}_2 = \hat{\boldsymbol{\Sigma}}_1$. This C-step therefor always finds a smaller determinant for the covariance until convergence (Rousseeuw & Driessen, 1999). However, there is no guarantee that this process leads to a global minimum. This is the reason the FAST-MCD algorithm makes multiple initial subsets H_1 and repeats the the procedure multiple times, keeping the subset with the lowest determinant after convergence.

3.1.3 One-step-ahead forecast

The process described above determines whether observations are likely to be originated from the within-fixation distribution. However due to anomalies and blinks in the data, not each observation where $d_t < \chi^2_{p,0.975}$ is necessarily a saccade. In order to alleviate this issue, the BIT algorithm uses a one-step-ahead forecast (Kumar, Winograd, Paepcke, & Klingner, 2007). For a potential saccade observation $\mathbf{z}_t$ this results in calculating the velocity from $t-1$ to $t+1$:

$$\Delta_2(\boldsymbol{z}_t) = (\boldsymbol{z}_{t+1} - \boldsymbol{z}_{t-1}). \tag{11}$$

This results in a small alteration of previously defined d_t:

$$v_t(\Delta_2(\boldsymbol{z}_t)) = \sqrt{(\Delta_2(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_0)' \hat{\boldsymbol{\Sigma}}_0^{-1} (\Delta_2(\boldsymbol{z}_t) - \hat{\boldsymbol{\mu}}_0)}, \tag{12}$$

This new variable is subjected to the same procedure $v_t < \chi^2_{p,0.975}$. If this statement is true, the one-step-ahead observation does not return back to within-fixation velocity. The point is then properly labelled as a saccade. In order for sequences of observations to be classified as fixations there must be at least three consecutive points that are classified as possible fixation points. An example of the final tolerance ellipse for a single eye is shown in the figure below.

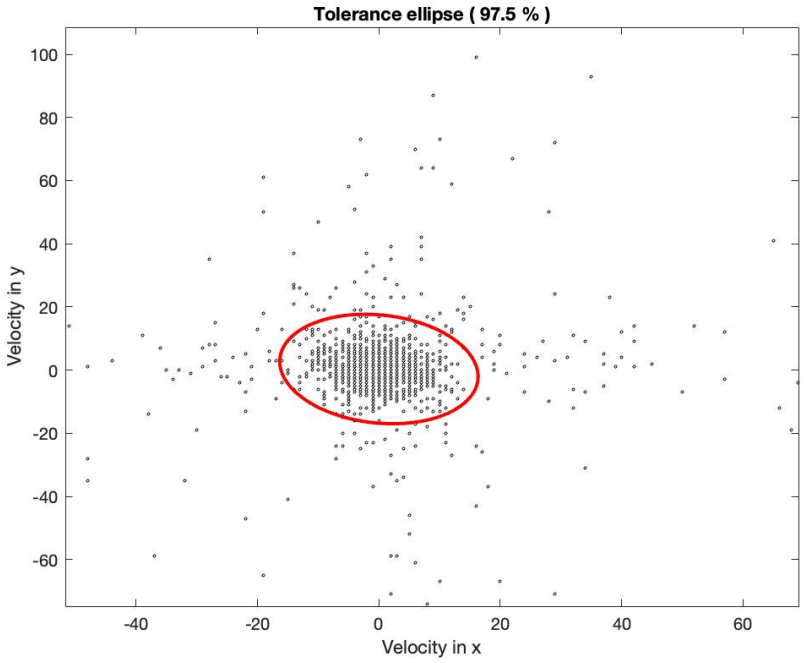


Figure 3: The tolerance ellipse of Participant 7. All points inside the ellipse are classified as potential parts of fixations.

3.2 Extension

The extensions of the BIT algorithm are divided in two sections. First, the extensions need to create two tolerance ellipses, one for the observations where the subject is gazing at text, and another for the observations where the subject is gazing at images. Secondly, the extensions use these ellipses to determine whether a single observation is part of a fixation or an outlier. In order to create the separate tolerance ellipses, the extension of the BIT algorithm includes a separate column of data to indicate whether a subject is gazing at an image. However, this labelling needs to be done robustly. In the following sections, I propose different ways of determining whether a single observation is gazing at an image or text through the dummy variable *Image* and I propose different methods of using the tolerance ellipses obtained from these labelled observations, to determine which observations are part of fixations.

In this section I use sequences of potential observations as an example where each digit reflects an individual observation of a window of 17ms. An individual digit is labelled with a 1 if the subject is gazing at an image and labelled with a 0 if the subject is gazing at text. A longer sequence of ones indicates a subject looking at an image for an extended period of time. An example looks as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}' \tag{13}$$

3.2.1 Uncertainty period

In the vector above I define two sections. The first part is a period of stability where a subject is gazing at an image, and the final part where a subject is gazing at text. However, the way the middle part is treated is a critical part of the extension. For now, this middle section is labelled the uncertainty period. This uncertainty period possibly occurs when a subject is gazing around the edges of an image labelled area, where the inherent within-fixation movement of the eye along with measurement uncertainty can put subsequent observations on either side of the border of the image labelled territory. In the research, different approaches for dealing with the uncertainty period are used and the differences between them analysed. The definition of when a uncertainty period starts or finishes is a result of confidence weights, which is explained next.

3.2.2 Confidence weights

In order to define uncertainty periods we assign a weight $w_{image,t}$ to each observation. The weight reflects a metric of confidence that a single observation's POR falls within the image region. An observation close the the center of the image region is set to $w_{image,t} = 1$, with an observation on the border of the region equal to $w_{image,t} = w_{text,t} = 0.5$. Here $w_{text,t}$ reflects the weight of an observation whose POR does not fall in the image region. It holds that $w_{image,t} + w_{text,t} = 1$ for all t. Confidence is constructed in such a way that it increases rapidly when the distance to the border becomes larger. A maximum scoring distance is set such that any observation with a larger distance from the border than this value is assigned a weight of 1. If the maximum scoring distance is 50 pixels for example, separate observations in the image region that are located 51 pixels and 150 pixels from the closest border to a text region are both assigned $w_{image,t} = 1$. This same scoring applies for the observations located in the text region. All observations with a smaller distance to the closest border than the maximum scoring distance are assigned a weight based on a steeper variation of the Sigmoid function.

$$S(d^*) = \frac{1}{1 + \exp(-3d^*)}, \tag{14}$$

where in this case d^* is the normalised distance to the border, where the upper bound is the maximum scoring distance. The weights are symmetric for both sides of the border. A ten pixel distance to an image border within the image region has the same weight $w_{image,t}$ as the weight $w_{text,t}$ with a ten pixel distance to an image border within the text region. This research deals with several different methods regarding confidence weights, which are explained in detail for each algorithm in section 3.3.

3.2.3 Tolerance ellipses

When the region labels have been assigned to the observations, the FAST-MCD algorithm is applied for each set of observations with a unique label. This means that the algorithm is run once for all observations with an image label and once for all observations a text label. This results in two different estimates of the originating multivariate normal distribution. The two different location estimates $\hat{\boldsymbol{\mu}}_{text}$ and $\hat{\boldsymbol{\mu}}_{image}$ and the two different scatter estimates $\hat{\boldsymbol{\Sigma}}_{text}$ and $\hat{\boldsymbol{\Sigma}}_{image}$ are the MCD estimates of the multivariate normal distribution of the observations within their respective region. These mean and scatter estimates determined by the FAST-MCD algorithm are then used to create the 97.5% tolerance ellipses with their center at the location estimate and their boundaries determined by the scatter estimate.

3.2.4 Determining fixations

Once the tolerance ellipses for the observations have been calculated through the FAST-MCD calculations, the algorithm determines whether an observation is likely to be part of a fixation based on these ellipses. First, the algorithm checks whether the observation lies within the boundaries of the 97.5% tolerance ellipse. If this is the case, it is automatically determined to be a potential fixation point. If the observation falls outside the boundaries of the tolerance ellipse, the one-step-ahead forecast is used to check whether the observation returns to the within-fixation velocity. If the observation does not return to the within-fixation velocity, it is labelled a saccade, with the observation being labelled as a potential fixation point otherwise. Fixations are then determined from the potential fixation points if there are at least 3 consecutive potential fixation points without being interrupted by a saccade.

This means that the differences in how fixations are classified is determined by the differences between the tolerance ellipses. In this research, different methods of determining which tolerance ellipse is used for an observation are proposed. The easiest method is to compare an observation with the label $Image = 1$ to the tolerance ellipse corresponding with the image region, this is the method applied in the Naive Labelling extension, which is described in further detail below. Another method is taking fractions of both tolerance ellipses based on the confidence weight of the observation. This allows for a single observation to be evaluated based on 20% of the image tolerance ellipse and 80% of the text tolerance ellipse when the confidence weights for that observation are $w_{image,t} = 0.2$ and $w_{text,t} = 0.8$ for example. Determining potential fixations through confidence weights is the main concept of the Weighted Labelling extension described below.

3.2.5 Restricting fixations behaviour

All suggested extensions described below place no restrictions on whether fixations are allowed to exist over multiple regions. This means that a single fixation can have observations both in the image and text regions. However, for all extensions described below, a restricted version is also created. These restricted versions limit a fixation to a single region. Technically this is equivalent to running the algorithm separately for all observations with differing labels. These versions are useful for fully separating fixations in the image region and fixations in the text region, which allows for analysis of fixation behaviour between the regions, such as image fixation duration versus text fixation duration. The restricted versions determine fixations in the same manner as the unrestricted versions described above by comparing the velocities of observations with the relevant tolerance ellipse and applying the one-step-ahead forecast for observations that lie outside the tolerance ellipse. However, in the restricted versions, if two consecutive observations have a differing image label, the fixation is automatically stopped. Note that this does allow multiple fixations to follow each other without being interrupted by a saccade, which is an unrealistic assumption. For this reason the restricted versions of the algorithms should only be used for comparing behaviour between the regions.

3.3 The extensions

Now that the principles of the uncertainty period, confidence weights and the *Image* label have been covered, multiple extensions are proposed that treat each one of these principles differently. The following section contains a detailed explanation of how each proposed algorithm attaches the *Image* label, uses uncertainty periods and in which way they utilise the tolerance ellipses for determining whether an observation is part of a fixation.

3.3.1 Naive Labelling

The naive labelling (NL) extension is the most basic version of the extension. This extension takes the average of the positions of the left and right eye and determines whether the average x position and the average y position both correspond to the POR of an image region. If this is the case, that observation gets the label $Image = 1$, with the value being $Image = 0$ otherwise. One tolerance ellipse is constructed for all observations with the label $Image = 1$ and another tolerance ellipse is constructed for all observations that are assigned the value $Image = 0$. In order to determine whether an observation is part of a fixation, they are compared with the tolerance ellipse that corresponds to their *Image* label. This algorithm does not utilise confidence weights.

3.3.2 Weighted Labelling

Weighted labelling (WL) utilises the confidence weights described in the earlier section. This algorithm also averages the positions of the eyes and determines whether the average position is in the POR of an image region and assigns the observation the *Image* label accordingly. Next to this, all observations are given a confidence weight. This weight is between 0 and 1 if the distance between the observation and the closest border between image and text regions is smaller than the maximum scoring distance.

In this extension uncertainty periods are defined to start with three consecutive observations which have a confidence score between 0 and 1. This period ends when three consecutive observations do have a score equal to either 0 or 1. Using three consecutive observations to define the start and end of the uncertainty period allows for saccades and blinks to be part of the period, it also matches the BIT rule that states that a fixation must consist of at minimum 3 consecutive observations. Every observation in a single uncertainty period is then assigned a value for their *Image* label, based on the average of the confidence scores of that uncertainty period. If one uncertainty period averages $\bar{w}_{text} = 0.7$, the entire period will receive the label $Image = 0$, since the average confidence is higher for the period being part of a text area than an image area. After these labels are assigned, two tolerance ellipses are constructed in the same way as mentioned in the NL extension.

In order to determine whether an observation is part of a fixation, the extension compares an observation to a linear combination of both tolerance ellipses, based on the observations confidence weights. This allows for a single observation to be compared to a mixture of the tolerance ellipses, in order to determine whether the observation is a possible part of a fixation. The means and covariance matrices of the tolerance ellipses are combined through

$$N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = w_{image,t} N(\boldsymbol{\mu}_{image}, \boldsymbol{\Sigma}_{image}) + w_{text,t} N(\boldsymbol{\mu}_{text}, \boldsymbol{\Sigma}_{text}) \tag{15}$$

.

3.3.3 Separate Uncertainty

The separate uncertainty (SU) algorithm works identically to the WL algorithm with the exception that the periods that are classified as uncertainty periods are not labelled into $Image = 1$ or $Image = 0$ based on their average confidence weights. These periods are treated as a separate multivariate normal distribution with its own location and scatter estimates. This method therefor results in three different tolerance ellipses per participant, one for the image region, one for the text region and one for the observations that are part of uncertainty periods. In

order to determine whether a single observation is possibly part of a fixation, this method compares an observation with the uncertainty period tolerance ellipse if the observation is labelled as being part of an uncertainty period and follows the naive labelling as described above otherwise.

3.3.4 Binocular Labelling

The binocular labelling algorithms (BL) is a binocular version of the NL algorithm. The positions of the POR of the eye are not averaged, but checked individually whether the POR belongs to an image region or not. If both eyes' POR belongs to an image region, the observation receives the label $Image = 1$. If only one eye has their POR in the image region, the observation receives the label $Image = 1/2$. The observation receives the label $Image = 0$ if neither of the eyes' POR is within the image region. Similarly to the SU algorithm, this algorithm results in three different tolerance ellipses per participant, with one ellipse for every possible value of the $Image$ variable. However, the confidence weights are not utilised in this method, the uncertainty periods are rather substituted by the $Image = 1/2$ regions. To determine whether observations are part of fixations, each observation is compared with the tolerance ellipse corresponding to its own value for the *Image* label. To summarise, a short overview of all the extensions used in this research are displayed in the table below.

Table 2: Overview of the different algorithms used in this research

Algorithm	Tolerance ellipses	Combines ellipses	Uses uncertainty periods
Original BIT (BIT)	1	N	N
Naive Labelling (NL)	2	N	N
Weighted Labelling (WL)	2	Y	Y
Separate Uncertainty (SU)	3	N	Y
Binocular Labelling (BL)	3	N	N*

*:The BL algorithms equivalent of an uncertainty period are the regions where $Image = 1/2$.

3.4 Comparing and analysing methods

The main point of interest in these algorithms is the classification of a series of observations being a fixation. To compare the extension to the original BIT algorithm I perform a qualitative analysis of observations that are differently classified as fixations between algorithms. This allows for insight in what effect the extended algorithm has on certain areas of observations. In order to determine whether these extensions improve the classification ability, they need to be tested on data that is pre-classified by experts in eye-tracking. As this research does not have access to this data, it is not included, instead focusing on differences between the two algorithms in fixation classification. Next to this qualitative analysis of observations, a sensitivity analysis is performed.

3.4.1 Sensitivity analysis

By using the restricted versions of each algorithm as described above, the algorithms are able to do separate analysis on fixations limited to each region. This research only utilises both the restricted and unrestricted versions of the algorithms, which are clearly differentiated in the results. When comparing the restricted versions to the regular extensions, we can analyse the effect of limiting fixations to a single region.

In order to determine how much impact the separation of tolerance ellipses has on the overall output of fixations, we apply a sensitivity analysis. First, we run our algorithms as intended and each participant will have two tolerance ellipses associated with them for each particular algorithm: the text tolerance ellipse and image tolerance ellipse. For our sensitivity analysis we reuse these ellipses to determine fixations for all observations of the respective participant. First, the text tolerance ellipse is used to determine fixations, then the image tolerance ellipse is used. If the separation of tolerance ellipses has a large impact on the labelling of fixations, then there is a substantial difference between the fixations determined by the text, image and BIT tolerance ellipse. If these methods all determine the exact same fixations for each participant, then splitting the tolerance ellipses in an image and text region has no added benefit.

4 Results

This section will cover the main differences between the algorithms used in this paper. The results regarding amount of fixations classified, along with fixation duration analysis are shown first, followed by the different tolerance ellipses produced by the different methods. Thirdly, some specific differences in fixations are analysed to get an understanding of the differences between these algorithms. Finally, a sensitivity analysis is performed where we look at the effect of using the the image and text ellipse individually to predict fixations for both regions. Throughout the results the sections are separated for the unrestricted versions, which allow single fixations to continue across regions, and restricted versions of the extensions, which limits a fixation to a single region. Throughout this research, the maximum scoring distance is defined as 40 pixels.

4.1 Fixations

In the following sections the amount of fixations detected by each algorithm are presented and analysed. This section first analyses the unrestricted algorithms, followed by the analysis of the restricted algorithms.

4.1.1 Unrestricted

The table below shows the amount of fixations classified by all unrestricted extensions compared to the original BIT algorithm.

Table 3: Amount of fixations classified by each unrestricted algorithm.

Algorithm	Number of fixations
Original BIT	8481
Naive Labelling	8471
Weighted Labelling	8468
Separate Uncertainty	8497
Binocular Labelling	8467

The table shows that all extensions classify a very similar amount of fixations compared to the original algorithm, with no extensions differing more than 16 fixations than the original. The only extension that classifies more fixations than the BIT algorithm is the SU algorithm. This is possibly caused by the fact that the tolerance ellipse for the uncertain region has lower velocity thresholds than the ellipse of the image region. This is illustrated in chapter 4.4. This smaller tolerance ellipse might classify observations with the label $Image = 1$ as outliers when they are part of an uncertainty period, whereas they would have been classified as a potential fixation within the image tolerance ellipse in the other extensions. Overall classifying less fixations than the BIT algorithm indicates longer fixation duration. This is possibly caused

by the large thresholds of all image tolerance ellipses compared to the BIT tolerance ellipse, which allows for more observations to be classified as potential fixations, lengthening the fixation duration.

4.1.2 Restricted

The table below shows the amount of fixations classified by the restricted versions of the extensions compared to the original BIT algorithm, which has no restricted version. The third and fourth column show the amount of fixations classified within the image or text regions. And the fifth column shows the amount of fixations classified by either the separately analysed uncertainty period, or the case in the binocular algorithm where $Image = 1/2$.

Table 4: Overview of the amount of fixations classified by each restricted algorithm.

algorithm	Total Fixations	Image Fixations	Text Fixations	Other Fixations
Original BIT	8481	NA	NA	NA
Naive Labelling	8564	2734	5830	NA
Weighted Labelling	8493	2692	5801	NA
Separate Uncertainty	8525	2180	5273	1072
Binocular Labelling	8935	2609	5793	533

The table shows that almost all restricted extensions classify more fixations than the original BIT algorithm, with the WL algorithm only classifying 8 more fixations. The BL algorithm is farthest off, by 454 more fixations classified. There is a much larger variation between the restricted extensions than the unrestricted ones. There is also a larger deviation from the original BIT algorithm, as compared to the unrestricted version. This seems to indicate that the restricted version perform less similar to the original BIT algorithm than the unrestricted versions. Interestingly, the restricted SU extension classifies approximately 500 fixations less than the restricted NL and WL extensions in both the image and text regions. This means that the fixations that are part of the uncertain region are approximately 50% fixations that were labelled as image fixations in the NL and WL extensions, and 50% text fixations. The restriction explains why the BL algorithm overshoots the BIT algorithm in this fashion. We see that 533 fixations from the BL algorithm come from the $Image = 1/2$ section of the data. Since these sequences of observations are short and usually not separate fixations, but fixations that are part of either the text or image regions it creates an overestimation of the amount of fixations in the data. For comparison, the total amount of observations that receive the label $Image = 1/2$ is 5325. These observations result in 533 fixations classifications, this is a fixation to observation rate of 10.0%. The data with the BIT algorithm includes 131630 observations, with a classified 8481 fixations. This is a fixation to observation rate of 6.4%. Another algorithm that possibly suffers from the restriction is the SU algorithm. This algorithm has a fixation to observation rate of 7.4%. While this rate is higher than the BIT algorithm, it does not appear to cause immediate issues when comparing the number of total fixations. However, we see

that it is not recommended to use a restricted version of algorithms that include very short sequences of observations within a single region, since this can cause the overestimation of fixations as seen in the restricted BL algorithm.

4.2 Identical fixations

This section covers the concept of identical fixations. A fixation from algorithm A is said to be identical to a fixation from algorithm B if all properties of the fixation (ID, timestamp, duration, average position) match. The section will analyse the percentage of identically classified fixations between algorithms, both in the unrestricted an restricted versions. A higher percentage compared to the BIT algorithm does not indicate a better performance, it is only a measure of how similar the extension classifies fixations compared to the BIT algorithm.

4.2.1 Unrestricted

The table below shows the percentage of identically classified fixations of the unrestricted algorithms compared to the BIT algorithm and the other extensions.

Table 5: Fraction of fixations found by the row algorithm that are also found in the column algorithm (in %).

Algorithm	BIT	NL	WL	SU	BL
BIT	100	91.3	91.8	90.9	89.7
NL	91.1	100	97.6	97.1	96.0
WL	91.6	97.6	100	95.8	95.2
SU	91.1	97.4	96.2	100	94.7
BL	89.5	95.9	95.2	94.4	100

The table shows that the BL extension performs most dissimilar to the other algorithms. This is likely caused by the fact that this is the only extension that is based on binocular information, which might cause a larger difference in the tolerance ellipses, than the differences caused by labelling of the image region. Furthermore, the table shows that the WL algorithm performs most similarly to the BIT algorithm, with 91.6% of the fixations found by the WL algorithm also being present in the fixations of the BIT algorithm. However, it appears that all extensions are very similar to each other. The lowest score of similarity between two extensions is 94.4% between the BL and SU algorithms. Which is still approximately 3 percentage points higher than any extension compared with the BIT algorithm. This implies that the method of labelling has a smaller effect on the classification of fixations, than the difference between BIT and any extension that uses labelling.

While the percentage of identical fixations is a decent estimate for similarity of performance between algorithms, it does not specify how similar those fixations themselves are. For this

reason we also examine the amount of identically classified observations, or the actual amount of observations that change from an outlier to a fixation or the other way around. This gives further detail on the similarity between these algorithms. The table below shows the pairwise comparisons between algorithms for the percentage of identically classified observations.

Table 6: Fraction of observations identically classified between algorithms (in %).

Algorithm	BIT	NL	WL	SU	BL
BIT	100	-	-	-	-
NL	99.4	100	-	-	-
WL	99.4	99.9	100	-	-
SU	99.4	99.9	99.8	100	-
BL	99.3	99.7	99.7	99.6	100

Once again we see that the BL extension performs most dissimilar to the other extensions and the BIT algorithm. As expected, the percentage of identically classified observations is higher than the amount of identically classified fixations. Overall, approximately 800-970 observations have a changed fixation status when using the extensions compared to the BIT algorithm. The change in identically classified fixations when using the extensions range from 710-890. This indicates that most changes in the fixations are caused by the change in a single or a couple of observations. The results from the table above also confirm that changing from BIT to any extension has a larger effect on the classification of fixations than changing between extensions.

4.2.2 Restricted

The table below shows the percentage of identically classified fixations by each restricted extension compared to the BIT algorithm.

Table 7: Overview of identically classified fixations by each algorithm compared to the BIT algorithm (in %).

Algorithm	Total fixations	Image fixations	Text fixations	Other fixations
BIT	100	NA	NA	NA
NL	74.8	67.1	77.3	NA
WL	77.5	73.1	80.0	NA
SU	71.4	64.3	75.7	55.5
BL	70.2	61.6	74.2	0

The results confirms our suspicion of the restricted BL algorithm, where 0 of the 533 fixations classified in the $Image = 1/2$ period match fixations in the BIT algorithm. This re-amplifies the issue with the restriction in the BL and possibly the SU algorithm. Furthermore we observe that the image regions have less identical fixations than their text counterpart in every algorithm. This is partly explained due to the text region ($Image = 0$) being the majority class in the data with roughly 70% depending on the labelling method. This means that the BIT's

tolerance ellipse matches the shape of the text tolerance ellipse more than that of the image ellipse. Overall the WL algorithm matches the BIT algorithm the most. This is likely due to fact that the WL algorithm classifies longer periods than the other algorithms as being part of text or image. There are fewer jumps from text to image and vice-versa in the WL algorithm, which limits the issues caused by the restriction. The next closest algorithm NL has more short periods being classified as either image or text regions, resulting in more switches between the two. Overall using the restricted versions of the algorithms result in a larger difference in classification of fixations compared to BIT than their unrestricted counterparts.

4.3 Fixation length

It is difficult to infer information about performance without professional labelling of fixations. However, we can use information from literature about fixations length to see what percentage of each algorithms fixation classifications fall within a reasonable window of fixation length. The majority of fixations do not last longer than 500ms (McConkie & Dyre, 2000). I therefore look at outliers of fixations that are longer than 500ms to give an idea of the proportion of unrealistic fixation classifications. This process is not done for unrealistically short fixations, since the algorithms have an inherent rule that a fixation must include at least three consecutive observations (51ms), which is within bounds of reasonable fixation duration (McConkie & Dyre, 2000). When analysing fixation length for the restricted versions, a separate analysis is performed to detect differences between image and text fixations.

4.3.1 Unrestricted

The table below shows information about average duration of fixations and percentage of large fixations classified for each algorithm in the research.

Table 8: Overview of fixation average fixations lengths (in ms) along with the percentage of fixations the last over 500ms for each algorithm.

Algorithm	Fixation length	Large fixations (in %)
BIT	212	3.1
NL	212	3.1
WL	212	3.1
SU	211	3.0
BL	213	3.1

The first columns of the table shows that all algorithms reach a similar average fixation length to the BIT algorithm. The SU extension has a lower average fixation length and also scores the lowest in percentage of large fixations classified. This average fixation length follows from the fact that the SU extension classified more fixations than any other algorithm. However, overall these differences are too small to give any indication of performance.

4.3.2 Restricted

The table for average fixation length across the different restricted extensions, along with the percentage of large fixations is shown below.

Table 9: Overview of fixation average fixations lengths (in ms) along with the percentage of fixations the last over 500ms for each algorithm.

Algorithm	Fixation length	Image fixation length	Text fixations length	Large fixations (in %)
BIT	212	NA	NA	3.1
NL	215	218	214	3.6
WL	212	216	211	3.3
SU	221	227	218	4.2
BL	213	217	211	4.1

The table shows that when observations are in the image region, the fixation length tends to be longer for all restricted algorithms. The final column shows us that all restricted algorithms create a larger percentage of fixations over 500ms than the BIT algorithm. This final column gives us a reasonable estimate of restricted algorithm performance. Of the adjusted algorithms the restricted SU and BL algorithms perform poorest with over 4% of the fixations being over 500ms. The restricted IL and WL algorithms are more competitive with the BIT algorithm in this regard. All restricted algorithms perform worse than their unrestricted counterparts. As speculated earlier, the restricted algorithms perform worse when there are more splits in the data, with the algorithms utilising three tolerance ellipses scoring worse than those utilising two. Not included in the table are the median fixation lengths, which are 183ms for all values combinations shown in the table. To visualise the lengths of the fixations, histograms of fixation duration are shown for the two best performing algorithms according to the table above: the IL and WL algorithms, compared to the BIT algorithm.

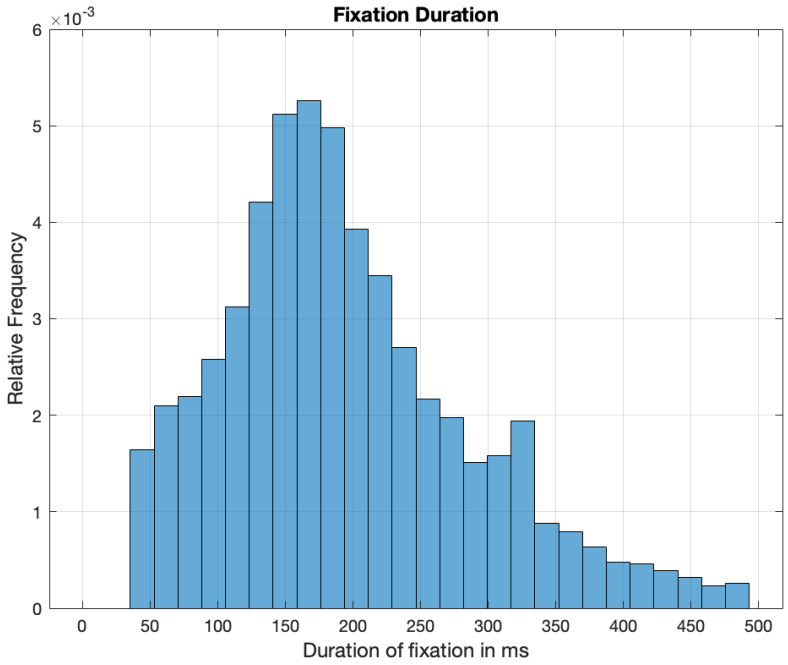
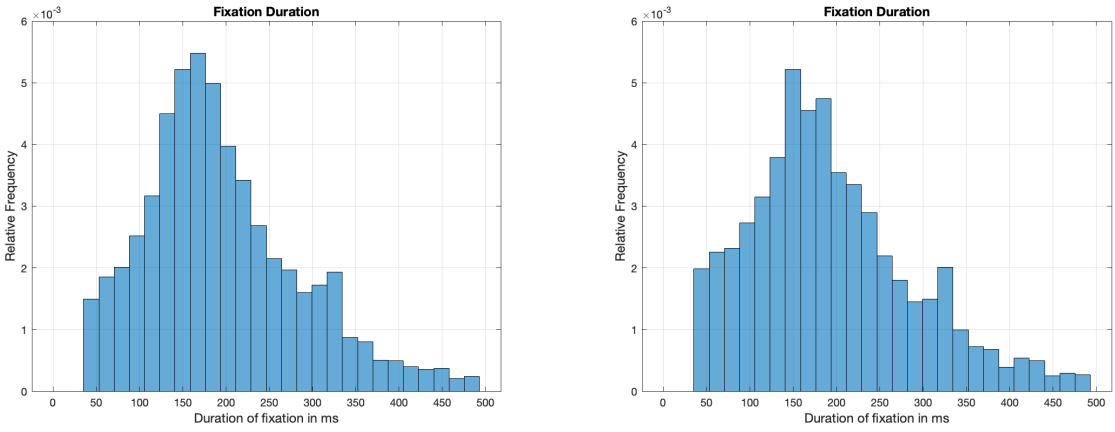
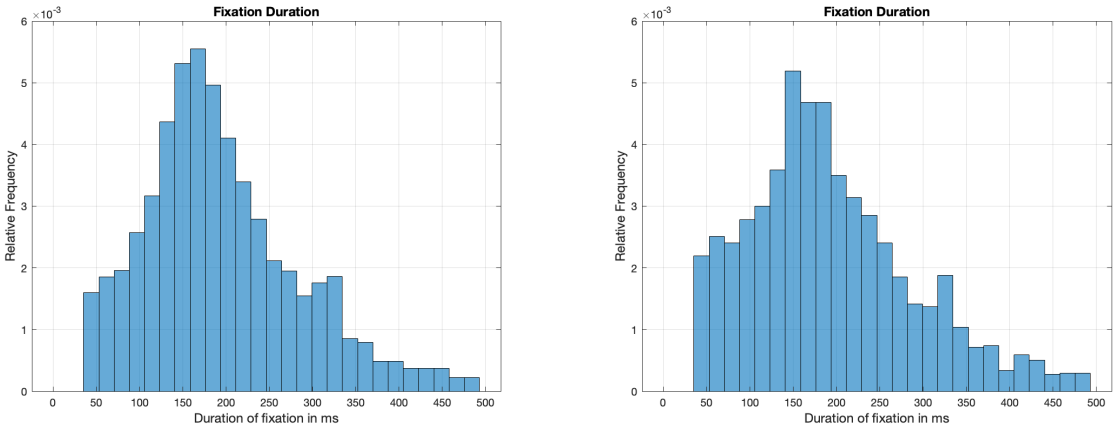


Figure 4: The relative frequency of fixation duration of the BIT algorithm.



(a) Text **(b)** Image

Figure 5: Relative frequency histograms for fixations lengths of the NL algorithm.



(a) Text **(b)** Image

Figure 6: Relative frequency histograms for fixations lengths of the WL algorithm.

These histograms show an important distinction between the text and image fixations. The bars left of the mode (<150ms) are relatively more frequent in image fixations than in text fixations. This behaviour is not reflected in the table with average fixation duration, which states that image fixations are longer on average. This indicates that the average fixation length for images is increased due to the amount of large fixations (>500ms) present in image fixations. Fixations with a duration of over 500ms are not displayed, since these observations are deemed to be incorrect classifications from the algorithms.

4.4 Tolerance ellipses

The tolerance ellipses are constructed separately per region for the extensions, with one general ellipse for the BIT algorithm. Below are figures with the ellipses for the BIT algorithm and the NL, WL and SU algorithms. These tolerance ellipses are constructed per participant and often vary across participants.

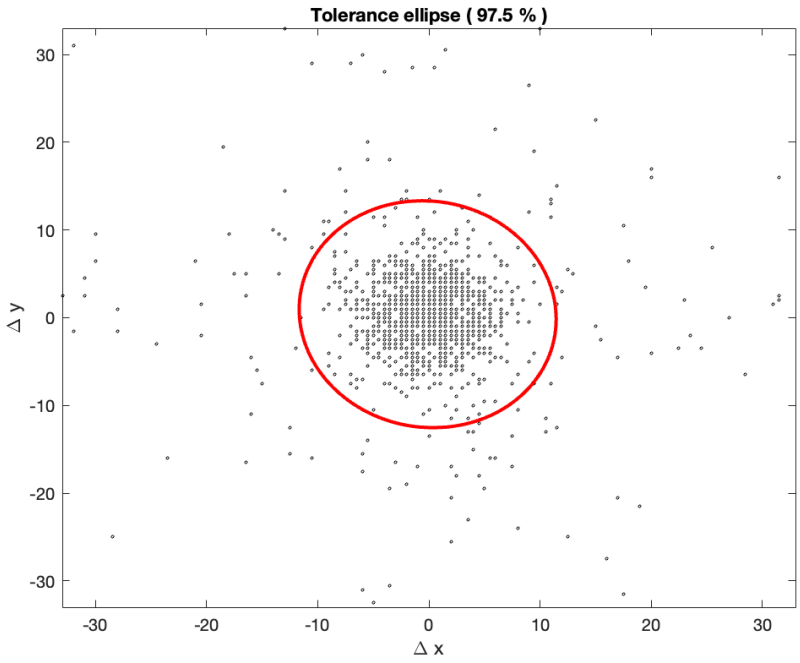
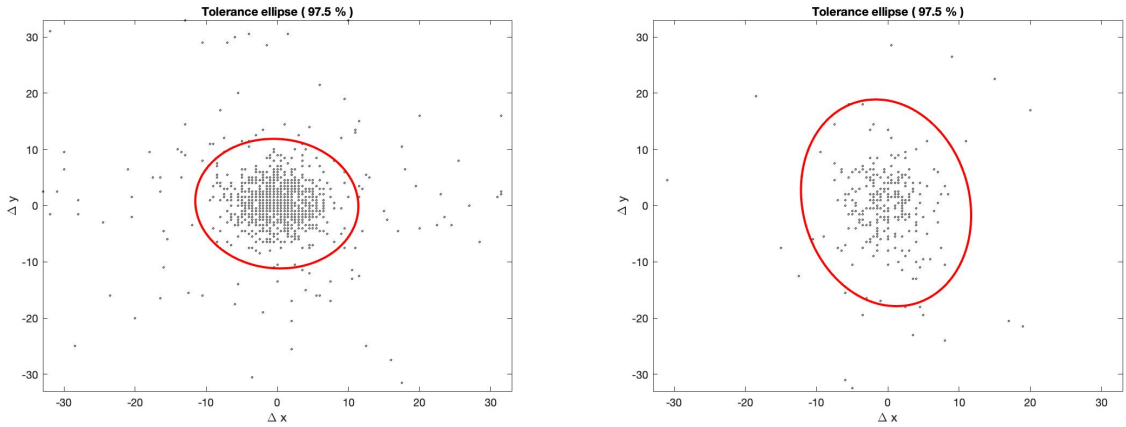
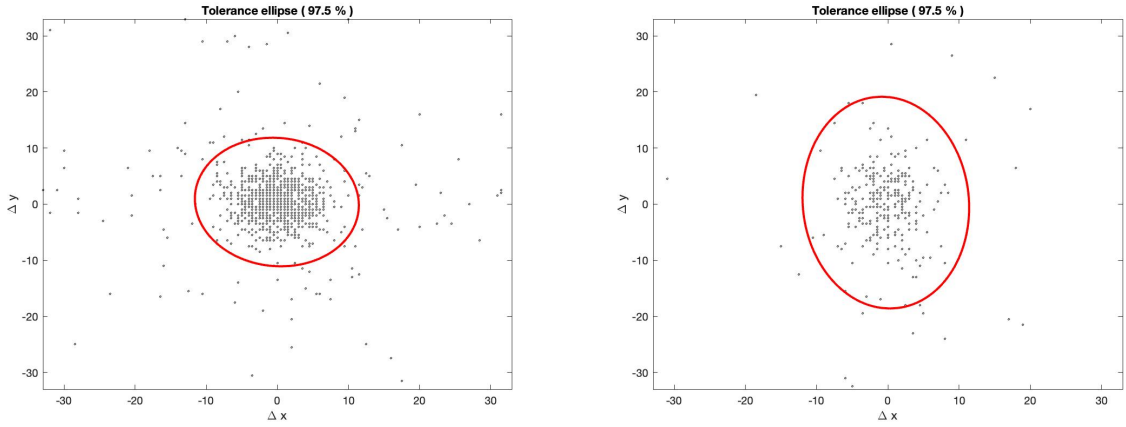


Figure 7: The tolerance ellipse for the BIT algorithm for participant 7.



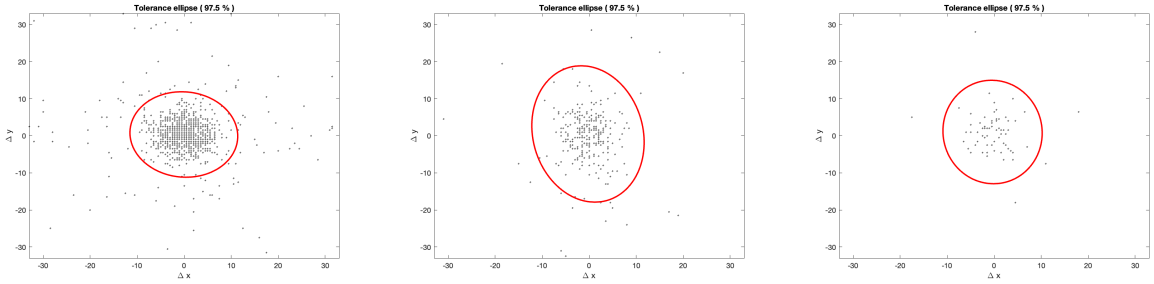
(a) Text **(b)** Image

Figure 8: Tolerance ellipses for the NL algorithm for participant 7.



(a) Text **(b)** Image

Figure 9: Tolerance ellipses for the WL algorithm for participant 7.



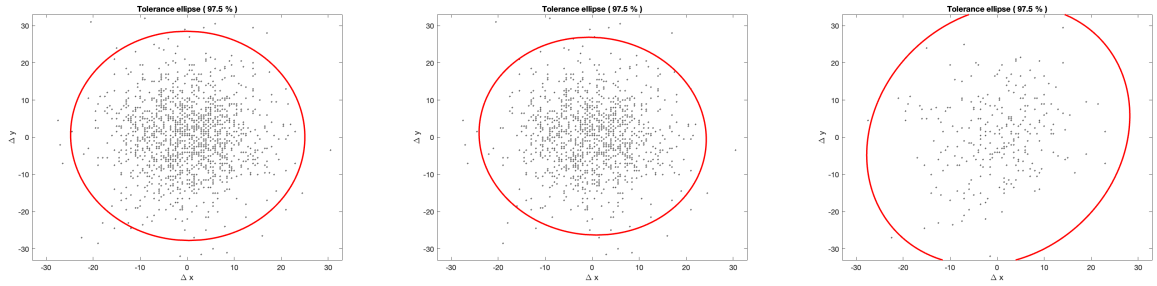
(a) SU Text **(b)** SU Image **(c)** SU Uncertain

Figure 10: Tolerance ellipses for the SU algorithm for participant 7.

For all algorithms, we see differences between the tolerance ellipses for text and image. The image tolerance ellipses are more spread, especially in the y-direction. This spread in the y-direction is to be expected. Since compared to reading text, looking at an image requires larger vertical eye movements. The shape of the BIT tolerance ellipse is very similar to that of the text ellipses. This is once again likely caused by the fact that the text observations are the majority class in the data, and thus influence the BIT ellipse more than the image class. The tolerance ellipse for the uncertain regions in the SU extension appears to be a mixture of the text and image ellipses. This is explained by the earlier finding that the fixations in the uncertain region are based on approximately 50% text and 50% image observations. While a clear distinction is visible between the tolerance ellipses, the difference across individuals is many times larger than the difference between text and image ellipses, with some individuals creating ellipses that range from -80 to 80 in both directions. This is not to say that this distinction between text and image is not important, but it should only be applied on individuals.

4.5 Outlier analysis

The detection of the longest fixation by the BIT algorithm is 10476ms by participant 9. It is noteworthy that this entire fixation takes place firmly in the text region of the slide. This particular fixation is also the final fixation of participant 9, with the two fixations leading up to it also both being over 1000ms, namely 1266ms and 1932ms. Interestingly, these three fixations all take place in the same region, with an average x position between 577 and 591 and an average y position between 526 and 597. Without the final fixation this participant has an average fixation time of 247ms, compared to 409ms if the final fixation is included. This is still higher than the average fixation length of 212ms from the BIT algorithm, but might be a more reasonable indication. All extensions combined the longest fixation with the preceding one to create a longer fixation of 11742ms, with the 1932ms fixation still appearing before that. The largest fixations correspond with a box of text that describes a toothbrush handle as plastic, which is nearly dead-center in the slide. Looking at the other fixations over 1000ms we see no suspicious pattern for locations on the screen, with all fixations varying across the entire slide. The tolerance ellipses for participant 9 for BIT, WL text and WL image respectively are shown below.



(a) BIT **(b)** WL Text **(c)** WL Image

Figure 11: Tolerance Ellipses for participant 9.

The first thing we notice, is that these tolerance ellipses are all approximately 2.5 times the diameter of the ones of the average participant. This large tolerance ellipse possibly explains why such long fixations are detected. Another interesting note is that the tolerance ellipse for text in the WL algorithm is noticeably smaller than the BIT ellipse.

4.6 Sensitivity analysis

When applying the tolerance ellipse for text determined by the WL algorithm for each individual participant, we see that 7937 fixations are identically labelled out of the 8481 BIT fixations, this is a percentage of 93.6%. When we repeat this process for the image tolerance ellipse, we see that 7351 out of 8481 fixations are identically labelled, or a percentage of 86.7%. We observe a clear difference between fixations using the tolerance ellipse of the BIT and the tol-

erance ellipses of text and image. We can conclude that separating the tolerance ellipses has a noticeable impact on the way fixations are determined. As seen in section 4.2, changing from the unrestricted to the restricted version of an extensions causes the percentage of identically labelled fixations to the BIT algorithm to drop by approximately 15 percentage points. We can also conclude that restricting the algorithms has a large impact on the classification of fixations. Finally, the usage of binocular data causes a drop in similarity of approximately 2 percentage points compared to both the BIT algorithm, and other extensions.

5 Conclusion

This paper compares multiple extensions of the BIT algorithm for eye fixation detection. The extensions are expanded versions of the original through their ability to determine whether a subject is gazing at an image or a piece of text within the context of a task. The four extensions all have different methods of determining whether a subject is gazing at an image or at text. These methods are based on a uncertainty period, which are periods where we the subject is gazing near a border between image and text regions. Through these models we aim to answer the following questions: Is there a significant difference in eye-movement when switching between images and text within a task? And secondly: What are the main differences in fixation classification between the original and the extended BIT model?

Overall, the unrestricted Separate Uncertainty extension was able to lower the amount of unrealistic fixations of above 500ms compared to the BIT algorithm, however this was a minimal reduction from 3.1% to 3.0%. All other unrestricted extensions scored identical to the BIT algorithm, with the restricted extensions all performing worse. All unrestricted extensions classify 89-92% of the same fixations as the original algorithm. The most similar performing extension to BIT is the Weighted Labelling extension with 91.6% of the classified fixations being identical to those of the BIT. The extensions showed a similar performance with respect to fixation lengths to the original model. The tolerance ellipses that show which observations are likely to be part of fixations change in all models when switching from an image to a text region. These changes, however, appear to only be relevant on an individual basis, since the variation of ellipses across individuals is multiple times larger than the variation between a single individuals text and image tolerance ellipse. All extensions were unable to eliminate the largest fixation classified by the BIT algorithm, with all unrestricted algorithms lengthening the largest fixation by 1266ms.

Sensitivity analysis showed that separating the image and text ellipses has a clear effect of up to 13% on the classification of fixations by the algorithms. The restriction placed on the restricted extensions that enforces them to only have one region (image or text) per fixation has an influence on the results, causing some of these extensions to perform very differently as compared to the BIT model.

I conclude that there is a difference in eye-movements when switching between images and text within a task for all extensions. These differences, however, are many times smaller than the differences that are caused by differences between individuals. I also conclude that the extended algorithms perform differently than the original, without a large difference in fixation classification. There is no notable difference between the extensions and the original in

terms of fixation length or classification of the group of longest fixations, with the exception of the longest fixation being lengthened by the extensions. We are not able to state anything about the quality of the performance of these models, except for the fact that the restricted versions scored worse on the percentage of unrealistic fixations classified. This research uses eye-averaged data for all but one of its extensions. This does not fully utilise the properties the BIT algorithm has to offer. Future extensions could have a larger focus on binocular data. Future research could apply these extensions to different concepts of distinguishable regions, such as colour areas versus black and white areas. These extensions are applicable for any separating of regions that is done manually. Future research could also experiment with these extensions on professionally labelled eye-tracking data. If they succeed in doing so, more conclusions can be drawn regarding the performance of the different models discussed in this research.

References

Engbert, R., & Mergenthaler, K. (2006). Microsaccades are triggered by low retinal image slip. *Proceedings of the National Academy of Sciences*, *103*(18), 7192–7197.

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford.

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics* (pp. 492–518). Springer.

Kumar, M., Winograd, T., Paepcke, A., & Klingner, J. (2007). *Gaze-enhanced user interface design* (Tech. Rep.). Stanford InfoLab.

Liechty, J., Pieters, R., & Wedel, M. (2003). Global and local covert visual attention: Evidence from a bayesian hidden markov model. *Psychometrika*, *68*(4), 519–541.

McConkie, G. W., & Dyre, B. P. (2000). Eye fixation durations in reading: Models of frequency distributions. In *Reading as a perceptual process* (pp. 683–700). Elsevier.

Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods*, *42*(1), 188–204.

Pison, G., Van Aelst, S., & Willems, G. (2002). Small sample corrections for lts and mcd. *Metrika*, *55*(1), 111–123.

Rousseeuw, P. J., & Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, *41*(3), 212–223.

Termsarasab, P., Thammongkolchai, T., Rucker, J. C., & Frucht, S. J. (2015). The diagnostic value of saccades in movement disorder patients: a practical guide and review. *Journal of clinical movement disorders*, *2*(1), 1–10.

Van der Lans, R., Wedel, M., & Pieters, R. (2011). Defining eye-fixation sequences across individuals and tasks: the binocular-individual threshold (bit) algorithm. *Behavior research methods*, *43*(1), 239–257.