

**Micro-Pattern Detection and Analysis in Gaze Data via  
Mathematical Optimization and Machine Learning**

Wen Liu

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

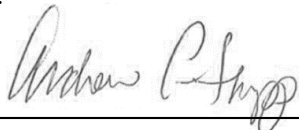
Degree of Doctor of Philosophy

in

Data Science

April 2019

APPROVED:



---

Professor Andrew C. Trapp (WPI, Advisor)



---

Professor Soussan Djanasbi (WPI, Co-Advisor)

---

Professor Jian Zou (WPI)



---

Professor W. Art Chaovalitwongse (University of Arkansas)

## Abstract

The use of eye tracking analysis to understand human behavior and cognition is increasingly prevalent in user experience research. Eye gaze data consists of a sequence of eye movement events, such as fixation and saccade, which can be used to analyze focus of attention and awareness under a variety of visual stimuli. The distribution of gaze points within individual fixations, which we call *micro-patterns*, has to date been largely unexplored. This work uses mathematical optimization and machine learning to explore micro-patterns in gaze data, and thereby improve the fundamental unit of analysis for attention and awareness in eye-tracking studies. The result is enhanced accuracy of location and level of attention intensity.

The primary research is to study micro-patterns in gaze data by developing fixation detection algorithms using data science technologies. *Fixation inner-density* (FID), introduced for the first time in this dissertation, measures the compactness of a fixation. It exhibits significant information about focused attention and effort. In Chapter 1, integer optimization and algorithmic techniques are combined to identify fixations in gaze point sequences by optimizing for inner-density. The computational results in Chapter 1 together with the experiments in Chapter 2, demonstrate that this approach, also known as the *FID filter*, outperforms methods used in existing commercial eye trackers in fixation refinement. Moreover, it has great potential to contribute to user experience research by providing better representation of attention and awareness, which is the fundamental unit of analysis in behavioral studies.

We further extend this research in two dimensions. The first extension, known as the *FID<sup>+</sup>* filter, advances the integer optimization techniques to identify *fixation outliers* in gaze point sequences. As introduced in Chapter 3, this enhances the FID filter by accounting for outlier sensitivity. The second extension is a set of experiments to explore the automated recommendation of the density intensity modulation parameter  $\alpha$  to the FID filter users. Chapter 4 discusses current findings from the experiments of recommending suitable  $\alpha$  levels on how two eye-tracking datasets were manually labeled, and experimental findings on recommending suitable  $\alpha$  levels.

These developments serve as fundamental building blocks for a real-time system for gaze fixation detection using inner-density. Such a system can provide instant and accurate gaze analysis, and thereby enable the ability to provide immediate feedback to the user. This may have significant implications and expand the application scope of eye tracking, and will be beneficial to Human Computer Interaction and behavioral research through the development of innovative and personalized user experiences.

## Acknowledgments

I would like to express my gratitude to my academic advisors, Prof. Andrew C. Trapp and Prof. Soussan Djamasbi. I want to especially thank Prof. Trapp for his support and kindness. I feel very grateful and lucky to have him in my life. He has become one of the most influential person to me. His guidance and encouragement make me grow as a professional researcher. It is my great honor to be his first PhD student. I would also like to express my thanks to Prof. Soussan Djamasbi, for her support and advice to my research and future career. In addition, I am very grateful to my dissertation committee members, Prof. Jian Zou and Prof. W. Art Chaovaitwongse, for their support and insights to my dissertation.

I am very thankful to the Data Science program at WPI, for providing support throughout my Ph.D. studies. I want to especially thank Prof. Elke A. Rundensteiner and Mrs. Mary Racicot, for providing financial support and taking care of student needs.

I would like to thank all the members in the UXDM lab at WPI. It is very cheerful and glad to have companions while doing research.

My special thanks go to my family members, who give my unconditional love and prayers. Lastly, I would like to thank my husband, Xiaotong, for always being generous, adorable, patient, and supportive throughout my life.

# Contents

<b>1</b>	<b>Identifying Fixations in Gaze Data via Inner-Density and Optimization</b>	<b>1</b>
1.1	Introduction . . . . .	2
1.2	Background . . . . .	3
1.2.1	Drawbacks of I-DT Filter . . . . .	4
1.2.2	Drawbacks of I-VT Filter . . . . .	4
1.3	Technical Development . . . . .	6
1.3.1	Fixation Identification: Formal Problem Description . . . . .	6
1.3.2	Three Mathematical Insights . . . . .	7
1.3.3	Mathematical Modeling . . . . .	9
1.3.4	Algorithm to Identify Densest Fixations . . . . .	13
1.4	Computational Experiments . . . . .	14
1.4.1	Datasets and Equipment . . . . .	14
1.4.2	Data Preprocessing . . . . .	15
1.4.3	Evaluation Metrics . . . . .	15
1.4.4	Computational Results and Discussion . . . . .	17
1.4.5	Polynomial-time Algorithm to Identify Single Fixation per Chunk . . . . .	21
1.5	Conclusions . . . . .	23
<b>2</b>	<b>Measuring Focused Attention Using Fixation Inner-Density</b>	<b>25</b>
2.1	Introduction . . . . .	25
2.2	Background . . . . .	26
2.3	Methodology . . . . .	27
2.4	Experimental Evaluation . . . . .	28
2.4.1	Dataset and Equipment . . . . .	28
2.4.2	Data Preprocessing . . . . .	29
2.4.3	Experimental Results . . . . .	29
2.4.4	Comparing I-VT and FID Filters for a Single Record . . . . .	30
2.4.5	Comparing I-VT and FID Filters for All 27 Remaining Records . . . . .	32
2.5	Conclusions . . . . .	35

<b>3</b>	<b>Outlier-Aware, Density-Based Gaze Fixation Identification</b>	<b>36</b>
3.1	Introduction . . . . .	36
3.2	Background . . . . .	38
3.3	Mathematical Developments . . . . .	40
3.3.1	Decomposition Principle . . . . .	40
3.3.2	FID <sup>+</sup> Filter: Detecting Fixation Outliers in Gaze Data . . . . .	41
3.3.3	Minimizing Square Area of Fixations with Outlier Sensitivity . . . . .	43
3.3.4	Deriving Lower Bounds on $r_f$ . . . . .	44
3.4	Computational Experiments . . . . .	49
3.4.1	Experimental Setup and Data Preprocessing . . . . .	49
3.4.2	Computational Results and Discussion . . . . .	50
3.5	Conclusions . . . . .	54
<b>4</b>	<b>Exploratory Data Analysis for Recommending <math>\alpha</math> to the FID Filter users</b>	<b>57</b>
4.1	Introduction . . . . .	57
4.2	Dataset and Equipment . . . . .	58
4.3	Data Labeling Toolbox . . . . .	59
4.4	Exploratory Data Analysis . . . . .	60
4.5	Predictive Modeling . . . . .	62
4.5.1	Training and Testing Datasets . . . . .	62
4.5.2	Feature Extraction . . . . .	62
4.5.3	Step One: Classification Model . . . . .	63
4.5.4	Step Two: Regression Model . . . . .	64
4.6	Findings and Discussions . . . . .	65
4.7	Conclusions . . . . .	66
<b>5</b>	<b>Conclusions and Future Work</b>	<b>68</b>

# Chapter 1

## Identifying Fixations in Gaze Data via Inner-Density and Optimization

Eye tracking is an increasingly common technology with a variety of practical uses. Eye tracking data, or gaze data, can be categorized into two main events: fixations represent focused eye movement, indicative of awareness and attention, whereas saccades are higher velocity movements that occur between fixation events. Common methods to identify fixations in gaze data can lack sensitivity to peripheral points, and may misrepresent positional and durational properties of fixations. To address these shortcomings, this chapter introduces the notion of inner-density for fixation identification, which concerns both the duration of the fixation, as well as the proximity of its constituent gaze points.

Moreover, this chapter demonstrates how to identify fixations in a sequence of gaze data by optimizing for inner-density, which is a representative of fixation *micro-patterns*. After decomposing the clustering of a temporal gaze data sequence into successive regions (chunks), we use nonlinear, linear 0–1 and second-order cone program optimization formulations to find the densest fixations within a given data chunk. Our approach is parametrized by a unique density intensity controller parameter  $\alpha$  that adjusts the degree of desired density, allowing decision makers to have fine-tuned control over the density in the process. We call the resulting algorithm as *fixation inner-density* (FID) filter. We show that our problem is fixed-parameter tractable, so we also develop a polynomial-time algorithm to find small number of fixations in data chunks. Computational experiments on real datasets demonstrate the efficiency of our optimization-based approach. Fixations identified through our approach exhibit greater density than existing methods, thereby enabling the refinement of key gaze metrics such as fixation duration and fixation center.

## 1.1 Introduction

Research interest in understanding human behavior and cognition via eye tracking techniques has existed for a long time. With the increasing availability of low-cost eye tracking devices, it is evaluated that eye tracking devices will become pervasive accessories for computers in the near future [1]. The basic function of an eye tracking device is collecting gaze data, which represent eye movement when a visual stimulus is presented. The study of gaze data is useful in many different areas, such as understanding of the human visual system [2], diagnosis of psychological disorders [3], analysis of marketing techniques [4], design of products [5], and web experience [6].

Precisely understanding the recorded gaze data plays a key role for eye movement behavior applications [7]. This understanding comes from the translation of raw, longitudinal gaze data into distinct eye-movement, or *oculomotor*, events. This process is known as *fixation* identification [8], and it separates gaze data into two primary event types: fixations and saccades. *Fixations* are pauses over informative regions of interest, where cognitive processing is believed to occur, whereas *saccades* are rapid movements between fixations, used to recenter the eye on a new location [8, 9]. Fixations are the primary unit of analysis for attention and awareness studies. Fixations characterize attention because they represent effort in maintaining a relatively stable gaze to take foveal snapshots of an object for subsequent processing by the brain [1].

To date, computational analysis has enabled a great deal of progress towards translating gaze data into fixations. Primary existing methods for identifying fixations use either *gaze location* (e.g., I-DT filter) or *velocity* metrics (e.g., I-VT filter). Methods based on the former typically use a constant area size as the threshold for grouping consecutive gaze points into a fixation, while the latter use a fixed velocity threshold to separate fixations from saccades. While these existing approaches are relatively simple to implement and generally effective, they can lead to issues with precision because they are prone to including points on the fringe of tolerance settings, thereby skewing summary fixation metrics (further discussed in Section 1.2).

Our work makes two novel contributions to address these shortcomings. The first is the identification of fixations via *inner-density*, which carries two characterizations of cognitive effort: the duration of a fixation, as well as its proximal compactness. It has been shown that fixation duration is a reliable measure of attention [1], and proximal compactness of individual gaze points in a fixation represent a person’s focused attention and increased levels of information processing [10]. Fixations with greater inner density tend to exclude peripheral gaze points, thereby improving the accuracy of traditional fixation metrics.

While there is great potential to use inner-density for refining gaze data, there are

no known studies that use the concept to identify fixations, let alone optimization-based approaches. Our second contribution is a *computational* approach to identify the *densest* fixations from gaze data, which we call the *fixation inner-density* (FID) filter. Given the impressive progress of modern optimization technology (for one such review in the context of data analysis see, e.g., [11]), exact methods that provide a performance guarantee on solution quality are now a reality; that is, given a dense fixation, optimization methods can prove *no denser fixation exists*. This is incredibly important when exact, rather than approximate, oculomotor event identification is desirable, or even essential.

## 1.2 Background

Gaze data has a particular structure, and must be reliably processed to generate meaningful information. Prior to proposing fixation *inner-density* and its associated optimization as a novel approach to measure information processing behavior, we review existing methods.

The process of fixation identification separates gaze data into distinct oculomotor events (e.g., fixations and saccades). The gaze data we consider results from user interaction with 2D static stimuli, e.g. visual computer displays, as a major focus of behavioral research is to understand user interaction with static screen based technologies. This gaze data is recorded in two dimensions  $(x, y)$  for every discrete time point  $t$ . Hence each 2D data point is an  $(x, y, t)$  triplet. Each time-series sequence  $\mathcal{S}$  of consecutive discrete  $(x, y, t)$  gaze data points can be computationally separated into constituent fixations. Common sampling rate frequencies range from 30 Hz to 300 Hz, though some eye tracking devices can record at levels exceeding 1,000 Hz [12]. Once gaze data has been computationally processed into its fundamental oculomotor events, each event can be characterized using summary statistics, for example the duration and center (centroid) of the event. Figure 1.1 depicts approximately 10,000 gaze points in a segment of a real, raw gaze data sequence in  $(x, y, t)$  space, which arises from a task of reading on a 2D static computer display stimulus. The problem of interest is to separate this gaze data into distinct fixations.

Two primary methods exist to analyze and process gaze data: those based on gaze-point *position*, such as the I-DT, and those based on gaze-point *velocity*, such as the I-VT for in-depth descriptions of these approaches, see, e.g., [8, 13]. It is widely accepted that all existing event detection methods have flaws [7]. This is due in part to the arbitrary and somewhat interpretive nature of classifying gaze data points into representative events. The author in [14] contends that the reason there are so many ways to identify fixations (clusters) is because the notion cannot precisely be defined; rather, it is in the eye of the



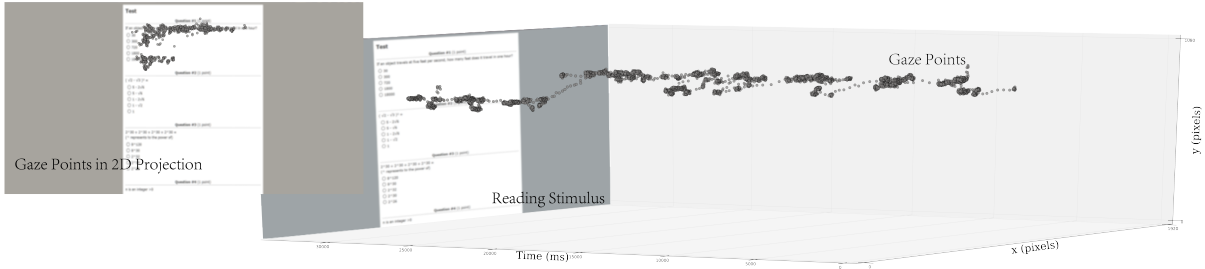


Figure 1.1: Raw  $(x, y, t)$  gaze data depicted in three dimensions, as recorded by a typical eye tracking device.

beholder. Even so, there are basic criteria, many used by existing approaches, that are suggestive for a group of points to be considered as a fixation.

### 1.2.1 Drawbacks of I-DT Filter

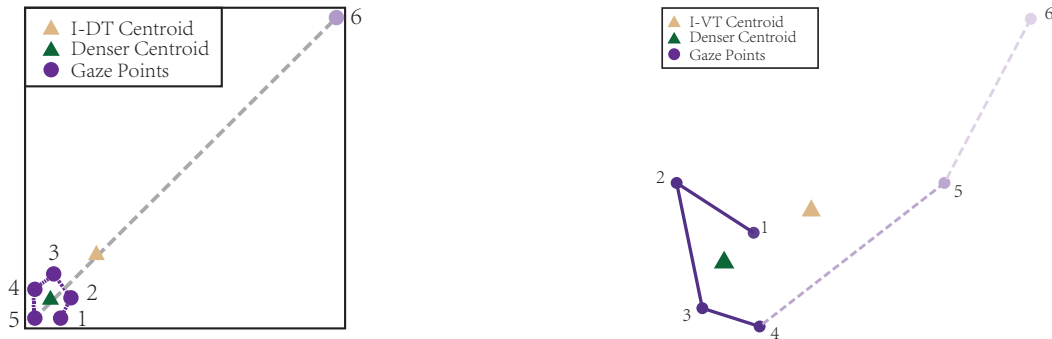
The I-DT is a well-known position-based approach. This algorithm separates gaze data using a predefined maximum dispersion threshold  $D$  together with a minimum duration. It uses a fixed-area window to construct fixations by sequentially adding points beyond a minimum duration, until the dispersion threshold is exceeded [8]. The I-DT can yield fairly accurate results, is rather straightforward to implement, and has favorable performance time. However, a significant drawback arises from the interaction with the threshold  $D$  and the dispersion metric it uses:

$$D(x, y) = D(x) + D(y) = [\max(x) - \min(x)] + [\max(y) - \min(y)]. \quad (1.1)$$

Figure (1.2a) illustrates some of the challenges with I-DT in assuming a simple, constant dispersion threshold  $D$ . As long as the  $D(x, y)$  measure does not exceed  $D$ , points are considered to belong to the same fixation. Figure (1.2a) raises significant doubts as to whether the sixth point belongs to the same fixation as the first five gaze points. This in turn can skew metrics such as the fixation duration and centroid, which are often used to assess user reaction to stimuli [15, 16].

### 1.2.2 Drawbacks of I-VT Filter

The I-VT algorithm may be the simplest of all fixation detection approaches, which sequentially categorizes each gaze point based on its point-to-point velocity. If the velocity meets or exceeds a velocity threshold  $V$ , it is identified as a saccade; below, the point belongs to a fixation [8]. I-VT is an elegant algorithm; as the authors in [8] discuss, it is a rather straightforward and robust approach, because the physical and physiological nature of the velocity profiles naturally separate data points into fixations or saccades. In



(a) The I-DT algorithm may misclassify gaze points under static dispersion threshold  $D$ . Whether to include the sixth point in the fixation, while technically within outer threshold  $D$ , is questionable.

(b) The I-VT algorithm may misclassify gaze points under constant velocity threshold  $V$ . The fifth and sixth points, while technically having velocities below threshold  $V$ , may not belong to the fixation.

Figure 1.2: Depicting some limitations of standard methods for fixation identification. For both the I-DT and I-VT algorithms, the center points (centroids) appear as lighter triangles, shifted to the upper right, as opposed to those of the denser fixation centroids, which are depicted with darker triangles and are more representative of the center of fixation of interest.

fact, the I-VT algorithm serves as the foundation for the fixation detection algorithms in major commercial eye tracking devices such as Tobii [17]. The I-VT algorithm features a simple implementation, efficient performance, and is fairly accurate.

Even so, the I-VT algorithm also has significant limitations. It essentially considers any consecutive group of points below a specified velocity threshold as a fixation. It then uses this grouping as a basis for summary statistics, such as the  $(x, y)$  centroid (by collapsing into a single point the individual  $x$  and  $y$  points according to their average values). Hence, the simplicity of the I-VT algorithm may result in misclassification, that is, points being classified as within the same fixation – when in reality they are distinct – because they do not strictly exceed the velocity threshold. As can be seen in Figure (1.2b), the inclusion of gaze points that are technically below the velocity threshold, but would not otherwise be included in a fixation, can skew important metrics such as the fixation duration and centroid. When considering the first four fixation points in Figure (1.2b), the centroid appears lower, and to the left, of where it appears when all six points are included in a fixation.

Although a few studies exist on enhancing the I-VT algorithm (see, e.g., [7, 18]), the aforementioned drawback remains detrimental for applications that demand precision. To resolve inherent discrepancies present in commonly used methods for fixation identi-

fixation, we propose the concept of *inner-density*, which refers to both the duration and concentration of the gaze points that form a fixation. In the next section we explain how we use inner-density to identify the densest fixations in a stream of temporal gaze data.

## 1.3 Technical Development

In this section we address the core challenge of *fixation identification* in gaze data. We begin with a formal problem description. We then highlight three unique insights to facilitate efficient solution of the problem, and proceed to introduce three mathematical programming formulations that identify fixations by optimizing for inner-density, together with an iterative algorithm.

### 1.3.1 Fixation Identification: Formal Problem Description

Fixation identification is the process of translating a longitudinal sequence of raw eye-movement data points into constituent fixation events and, thereby, the saccadic events between them [8]. We are unaware of any formal characterization of the *fixation identification* problem, though a related problem of *sequence segmentation* is discussed in [19], from which we adapt some notation.

Formally, we consider a raw time-series sequence  $\mathcal{S}$  of  $\mathcal{T}$   $d$ -dimensional gaze points, so that  $\mathcal{S} = \{t_1, \dots, t_{\mathcal{T}}\}$ . Let  $S_{\mathcal{T}}$  denote all such sequences of length  $\mathcal{T}$ . We seek to form  $\mathcal{F}$  fixations from these  $\mathcal{T}$  gaze points, with  $\mathcal{F}$  known a priori. An important consideration is to determine which points belong to fixations; some should not be included as they are saccade points, or possibly some other noise. Points that do form fixations must be consecutive in time, and together should be of a minimum length to have meaning with respect to cognitive processing. At a fixed sampling frequency, this is equivalent to stating that every fixation must contain a minimum number of points  $\mathcal{N}$ . Hence, the  $\mathcal{F}$  formed fixations constitute *segments* of the gaze sequence  $\mathcal{S}$  that are mutually exclusive, and of a sufficient minimum length. Of particular interest to us are *dense* fixations, which we will further qualify.

An  $\mathcal{F}$ -segmentation  $F$  of  $\mathcal{S}$  can be uniquely represented by  $\mathcal{F}$  pairs of fixation “segment” breakpoints. That is,  $F = \{(f_1, f_2), \dots, (f_{2\mathcal{F}-1}, f_{2\mathcal{F}})\}$ , with  $f_i \in \mathcal{S}$ . These pairs of breakpoints denote the fixation points in  $F$  through the respective intervals  $[f_{2j-1}, f_{2j}]$ ,  $j = 1, \dots, \mathcal{F}$ . Hence fixation  $j$  contains  $f_{2j} - f_{2j-1} + 1$  gaze points, which must meet or exceed  $\mathcal{N}$  for information processing to occur, so that  $f_{2j-1} + \mathcal{N} - 1 \leq f_{2j}$ ,  $j = 1, \dots, \mathcal{F}$ .

Let  $\mathcal{S}_{\mathcal{T}}$  denote all possible segmentations of gaze sequences of length  $\mathcal{T}$ , and let  $\mathcal{S}_{\mathcal{T}, \mathcal{F}, \mathcal{N}}$  denote all possible segmentations of sequences of length  $\mathcal{T}$  into  $\mathcal{F}$  fixation “segments” of

length  $\mathcal{N}$  or greater. Of particular interest is to minimize error criterion  $E : S_{\mathcal{T}} \times \mathcal{S}_{\mathcal{T}} \mapsto \mathbb{R}$  that assesses the quality of the formed fixations. Specifically,  $E$  should characterize two density-related aspects: fixation duration (a relatively *large number of* gaze points) and compactness (gaze points in *close proximity*).

For sequence  $\mathcal{S}$  and error function  $E$ , we define the optimal  $\mathcal{F}$ -segmentation  $F$  of  $\mathcal{S}$  as:

$$F_{opt}(\mathcal{S}, \mathcal{F}) = \arg \min_{F \in \mathcal{S}_{\mathcal{T}, \mathcal{F}, \mathcal{N}}} E(\mathcal{S}, F), \quad (1.2)$$

that is,  $F_{opt}$  is a grouping of  $\mathcal{S}$  into  $\mathcal{F}$  fixations that minimizes the function  $E(\mathcal{S}, F)$ .

**Problem 1 *Fixation Identification.*** *Given a raw longitudinal gaze sequence  $\mathcal{S}$  containing  $\mathcal{T}$  total time points, integer values  $\mathcal{F}$  and  $\mathcal{N}$  respectively denoting the number of fixations and minimum number of points, together with error function  $E$ , identify  $F_{opt}(\mathcal{S}, \mathcal{F})$ .*

As it turns out, this problem has a very large number of possible segmentations.

A related problem, *sequence segmentation*, is also concerned with optimal segmentation of time series sequences of data [19, 20]. They too consider minimizing an error criterion, for example distance from the center of the sequence. However a key distinction is that in the *sequence segmentation* problem, *all points* must be used to form relevant segments (clusters). On the contrary, the *fixation identification* problem forms fixations with *only the most salient time points* – that is, there are data points in the gaze sequence that should not be included in any fixation. A dynamic programming algorithm is presented in [19] to solve the *sequence segmentation* problem in  $\mathcal{O}(\mathcal{T}^3\mathcal{F})$  time, and it is further reduced to  $\mathcal{O}(\mathcal{T}^2\mathcal{F})$  time through a series of clever algorithmic improvements.

While a dynamic program similar to that of [19] also exists for the *fixation identification* problem, it has  $\mathcal{O}(\mathcal{T}^3\mathcal{F})$  complexity due to the need to process the assignment of points to fixations as well as to intervals between fixations, and unfortunately it becomes prohibitive to solve for even modest sizes of the fixation identification problem (indeed, [19] similarly notes “...*cubic complexity makes the dynamic programming algorithm prohibitive to use in practice.*”). This suggests alternative solution approaches, which are discussed in Section 1.4.5 are necessary that further exploit the structure of the *fixation identification* problem.

### 1.3.2 Three Mathematical Insights

We next highlight insights that enable us to develop an algorithmic approach to identify the densest fixations in  $\mathcal{S}$ .

## Decomposition Principle: Saccades Separate Fixations

A gaze sequence  $\mathcal{S}$  contains a large number of  $(x, y)$  points over time. Common lengths of gaze data sequences are in the tens to hundreds of seconds. For frequencies of 30 Hz to 300 Hz,  $\mathcal{S}$  can contain anywhere from several hundred, to hundreds of thousands of gaze points, and may contain hundreds if not thousands of fixations. For such realistic data instances, the fixation identification problem is prohibitive for even a moderate number of fixations, as proving the optimality of clusters on large datasets is known to be computationally demanding [21–23].

An alternative perspective leverages the specific structure of the sequence  $\mathcal{S}$ . Fixations must occur over temporally consecutive gaze points. Hence, any point that is identified as saccadic (e.g., by the I-VT filter) is a separator of fixations. Moreover, any small number of consecutive points may be removed if they are below a reasonable lower threshold for information processing to occur (similar to the I-DT filter). By removing these two types of gaze points, the gaze sequence  $\mathcal{S}$  becomes a collection of disjoint sets, or chunks, of gaze points where fixations may occur – that is, there are no saccadic points, and each chunk contains at least a minimum number of gaze points to be considered a fixation. Such a process separates  $\mathcal{S}$  into  $\mathcal{K}$  chunks of potential fixation points  $\mathcal{C}^k$ ,  $k = 1, \dots, \mathcal{K}$ . In particular,  $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset$ ,  $1 \leq i < j \leq \mathcal{K}$ , and  $\cup_{k=1, \dots, \mathcal{K}} \mathcal{C}^k \subseteq \mathcal{S}$ . Each of these chunks can subsequently be explored, independently, for (dense) fixations.

## Fixations Contain Consecutive Points in Time

There are fundamental differences between clustering temporal versus non-temporal data. In particular, fixations must adhere to temporal restrictions, which represents an extra condition for typical (atemporal) clustering tasks. Once a fixation begins, the included points must be consecutive in time, until the fixation ends. Stated another way, a fixation may conclude only once in a given sequence of gaze points. If this were not the case, fixations that occur in the same proximity, but separated over distinct periods of time, may be considered as a single fixation. Moreover, saccadic points that collect over time in the same region could also be incorrectly classified as a fixation [24]. To facilitate the ensuing discussion, define  $\mathcal{TF}$  binary variables  $z$ , with  $z_{tf} = 1$  if gaze point  $t$  is included in fixation  $f$ , and 0 otherwise.

$$\sum_{j=t+1}^{\mathcal{T}} z_{jf} \leq \mathcal{T}(1 - z_{tf} + z_{t+1,f}), \quad t = 1, \dots, \mathcal{T} - 1; \quad f = 1, \dots, \mathcal{F} \quad (1.3)$$

Time consistency constraint set (1.3) ensures that every fixation  $f$  has only consecutive gaze points and terminates at most once. For a fixation  $f$  starting at time point  $p$  and

concluding at  $q$ , the constraint set in (1.3) ensures in a linear fashion that  $z_{t,f} = 0$ ,  $z_{t,f} = 1$ , and  $z_{t,f} = 0$ . Moreover, this is accomplished with  $\mathcal{TF} - \mathcal{F}$  additional constraints, and no new variables.

### Controlling Inner-Density of Fixations

Given that fixation identification is somewhat subjective in nature, all automated classification methods require some interpretation. Fixations properties can fluctuate as the task and stimulus vary. To account for this, we incorporate a nonnegative parameter  $\alpha$  that acts to balance the tradeoff between the inclusion of additional gaze points and the spatial concentration of gaze points within fixations. This is done by incorporating the following term in the objective function:

$$\sum_{f=1}^{\mathcal{F}} \sum_{t=1}^{\mathcal{T}} \alpha(1 - z_{tf}), \quad (1.4)$$

where larger  $\alpha$  values provide greater incentive (that is, greater penalty) to include additional fixation points, at the expense of spatial proximity. Fixation inner-density can thereby be controlled by adjusting the level of  $\alpha$ . As  $\alpha$  increases, there is additional incentive to cluster points, with  $\alpha \rightarrow \infty$  tantamount to clustering all points (as in [19]).

### 1.3.3 Mathematical Modeling

We next present three optimization-based formulations that make use of these three key insights to identify fixations in gaze data chunks by optimizing for density. The first formulation bears some resemblance to a clustering approach proposed by [25], which has the advantage of finding 2D fixations with no strong regard for their shape. The formulation is nonlinear and requires linearization to solve efficiently. The second is an original, linear formulation that we develop, and has a related goal of bounding fixations with a square box of minimal diameter. The third is a second-order cone programming formulation for using circular bounding regions in fixation identification. We note that the following mathematical programming formulations are valid for any values of  $\mathcal{T}$  and  $\mathcal{F}$ , notably including smaller values that arise from the output of the decomposition principle described in Section 1.3.2, i.e. a single chunk  $\mathcal{C}^k$ .

#### MINLP Formulation: Minimize Average Intra-Fixation Sum of Distances

The main idea of this formulation is to ensure that fixations are constructed by minimizing the average intra-fixation sum of distances. Whereas every point must have a cluster

assignment in [25], in our formulation we enable gaze points to be selected for a fixation only when it improves the objective of optimizing the density-based metric – it is not necessary to include every data point in a given chunk. To offset the tendency to select fixations of minimum duration, we incorporate the idea in (1.4) to balance the tradeoff between highly compact clusters and non-inclusion. Our formulation uses values  $d_{ij}$  as the Euclidean distances between two data points  $i$  and  $j$ ,  $i < j$ , and  $\mathcal{N}$  is the minimum number of gaze points that could reasonably constitute a fixation.

$$\text{minimize } \sum_{f=1}^{\mathcal{F}} \left[ \frac{\sum_{i=1}^{\mathcal{T}-1} \sum_{j=i+1}^{\mathcal{T}} d_{ij} z_{if} z_{jf}}{\sum_{t=1}^{\mathcal{T}} z_{tf}} + \alpha \sum_{t=1}^{\mathcal{T}} (1 - z_{tf}) \right] \quad (1.5a)$$

$$\text{subject to } \sum_{f=1}^{\mathcal{F}} z_{tf} \leq 1, \quad t = 1, \dots, \mathcal{T}, \quad (1.5b)$$

$$\sum_{t=1}^{\mathcal{T}} z_{tf} \geq \mathcal{N}, \quad f = 1, \dots, \mathcal{F}, \quad (1.5c)$$

$$\sum_{j=t+1}^{\mathcal{T}} z_{jf} \leq \mathcal{T}(1 - z_{tf} + z_{t+1,f}), \quad t = 1, \dots, \mathcal{T} - 1; \quad f = 1, \dots, \mathcal{F}, \quad (1.5d)$$

$$z_{tf} \in \{0, 1\}, \quad t = 1, \dots, \mathcal{T}, \quad f = 1, \dots, \mathcal{F}. \quad (1.5e)$$

Constraint set (1.5b) ensures that gaze points are assigned to at most one fixation. Constraint set (1.5c) ensures a fixation contains at least  $\mathcal{N}$  points, and as per 1.3, constraint set (1.5d) ensures a fixation concludes at most once. Objective function (1.5a) contains two terms, one resembling the objective of [25], and a second that incentivizes inclusion of gaze points into fixations. Rather than  $d_{ij}^2$  as in [25], we use a simpler objective term of  $d_{ij}$  (this effect can be offset by adjusting the level of  $\alpha$ ). This formulation has  $\mathcal{T}\mathcal{F}$  binary variables and  $\mathcal{T}\mathcal{F} + \mathcal{T}$  linear constraints. The specific instance with  $\alpha$  very large and  $\mathcal{N} = 1$  yields a model that can solve the sequence segmentation problem of [19].

The first term of the objective function is nonlinear and fractional. In addition to containing the ratio of variable terms, it has a bilinear product component  $z_{if}z_{jf}$  in the numerator. This bilinear term can be linearized by introducing variables  $y_{ijf} \in \mathbb{R}_+$  equal to the product of  $z_{if}z_{jf}$ , enforced implicitly via the following three constraint sets:

$$y_{ijf} \leq z_{if}, \quad i = 1, \dots, \mathcal{T} - 1; \quad j = i + 1, \dots, \mathcal{T}; \quad f = 1, \dots, \mathcal{F}, \quad (1.6a)$$

$$y_{ijf} \leq z_{jf}, \quad i = 1, \dots, \mathcal{T} - 1; \quad j = i + 1, \dots, \mathcal{T}; \quad f = 1, \dots, \mathcal{F}, \quad (1.6b)$$

$$y_{ijf} \geq z_{if} + z_{jf} - 1, \quad i = 1, \dots, \mathcal{T} - 1; \quad j = i + 1, \dots, \mathcal{T}; \quad f = 1, \dots, \mathcal{F}. \quad (1.6c)$$

The remaining nonlinear fractional term of the objective can be linearized through an approach similar to [21, 26]. Define  $u_f = \frac{1}{\sum_{t=1}^{\mathcal{T}} z_{tf}}$ ,  $f = 1, \dots, \mathcal{F}$ . Continuous variable  $u_f$  has a lower bound of  $1/\mathcal{T}$  and, from (1.5c), an upper bound of  $1/\mathcal{N}$ . This gives a new objective function of:

$$\sum_{f=1}^{\mathcal{F}} \sum_{i=1}^{\mathcal{T}-1} \sum_{j=i+1}^{\mathcal{T}} d_{ij} y_{ijf} u_f, \quad (1.7)$$

which remains nonlinear. As  $y_{ijf}$  is binary and  $u_f$  is a bounded continuous variable, this product can be further linearized in a manner similar to (1.6a)–(1.6c). Define continuous variable  $v_{ijf}$  to be the product of  $y_{ijf} u_f$ . We can enforce this relationship implicitly through the following four constraint sets:

$$v_{ijf} \leq \frac{1}{\mathcal{N}} y_{ijf}, \quad i = 1, \dots, \mathcal{T} - 1; j = i + 1, \dots, \mathcal{T}; f = 1, \dots, \mathcal{F}, \quad (1.8a)$$

$$v_{ijf} \geq \frac{1}{\mathcal{T}} y_{ijf}, \quad i = 1, \dots, \mathcal{T} - 1; j = i + 1, \dots, \mathcal{T}; f = 1, \dots, \mathcal{F}, \quad (1.8b)$$

$$v_{ijf} \leq u_f - \frac{1}{\mathcal{T}} (1 - y_{ijf}), \quad i = 1, \dots, \mathcal{T} - 1; j = i + 1, \dots, \mathcal{T}; f = 1, \dots, \mathcal{F}, \quad (1.8c)$$

$$v_{ijf} \geq u_f - \frac{1}{\mathcal{N}} (1 - y_{ijf}), \quad i = 1, \dots, \mathcal{T} - 1; j = i + 1, \dots, \mathcal{T}; f = 1, \dots, \mathcal{F}. \quad (1.8d)$$

Lastly, it is important to ensure that  $u_f$  is indeed the reciprocal of  $\sum_{t=1}^{\mathcal{T}} z_{tf}$ . Akin to [21], we can restrict the sum of the variable  $v_{ijf}$  over all  $i, j$ ,  $i < j$  pairs. Suppose  $\sum_{t=1}^{\mathcal{T}} z_{tf} = \mathcal{P}$ . Then it is not difficult to show that  $\sum_{i=1}^{\mathcal{T}-1} \sum_{j=i+1}^{\mathcal{T}} y_{ijf} = \frac{\mathcal{P} \cdot (\mathcal{P}-1)}{2}$ . Hence  $\sum_{i=1}^{\mathcal{T}-1} \sum_{j=i+1}^{\mathcal{T}} v_{ijf} = \frac{\mathcal{P} \cdot (\mathcal{P}-1)}{2} / \mathcal{P} = \frac{\mathcal{P}-1}{2}$ . Rewriting this expression yields:

$$2 \sum_{i=1}^{\mathcal{T}-1} \sum_{j=i+1}^{\mathcal{T}} v_{ijf} - \sum_{t=1}^{\mathcal{T}} z_{tf} = -1, \quad f = 1, \dots, \mathcal{F}. \quad (1.9)$$

The final, linearized reformulation is:

$$\text{minimize} \quad \sum_{f=1}^{\mathcal{F}} \left[ \sum_{i=1}^{\mathcal{T}-1} \sum_{j=i+1}^{\mathcal{T}} d_{ij} v_{ijf} + \alpha \sum_{t=1}^{\mathcal{T}} (1 - z_{tf}) \right], \quad (1.10a)$$

subject to (1.5b), (1.5c), (1.5d),

$$(1.6a), (1.6b), (1.6c), (1.8a), (1.8b), (1.8c), (1.8d), (1.9), \quad (1.10b)$$

$$\frac{1}{\mathcal{T}} \leq u_f \leq \frac{1}{\mathcal{N}}, \quad f = 1, \dots, \mathcal{F}, \quad (1.10c)$$

$$0 \leq v_{ijf} \leq \frac{1}{\mathcal{N}}, \quad i = 1, \dots, \mathcal{T} - 1; j = i + 1, \dots, \mathcal{T}; f = 1, \dots, \mathcal{F}, \quad (1.10d)$$

$$z_{tf} \in \{0, 1\}, \quad t = 1, \dots, \mathcal{T}, \quad f = 1, \dots, \mathcal{F}, \quad (1.10e)$$

$$y_{ijf} \in \{0, 1\}, \quad i = 1, \dots, \mathcal{T} - 1; j = i + 1, \dots, \mathcal{T}; f = 1, \dots, \mathcal{F}. \quad (1.10f)$$



### MIP Formulation: Minimize Square Area of Fixations

We now present our second formulation for finding dense fixations. It attempts to balance enveloping the largest number of points with a 2D square of minimal area, as measured by the side length  $r$ . As in the first formulation, the model is parametrized by the expression described in (1.4).

$$\text{minimize } \sum_{f=1}^{\mathcal{F}} \left[ r_f + \alpha \sum_{t=1}^{\mathcal{T}} (1 - z_{tf}) \right], \quad (1.11a)$$

$$\text{subject to } (1.5b), (1.5c), (1.5d), \quad (1.11b)$$

$$x_f - r_f - \mathcal{M}_x(1 - z_{tf}) \leq x^t \leq x_f + r_f + \mathcal{M}_x(1 - z_{tf}), \quad t = 1, \dots, \mathcal{T}, \quad (1.11c)$$

$$y_f - r_f - \mathcal{M}_y(1 - z_{tf}) \leq y^t \leq y_f + r_f + \mathcal{M}_y(1 - z_{tf}), \quad t = 1, \dots, \mathcal{T}, \quad (1.11d)$$

$$l_x \leq x_f \leq u_x; l_y \leq y_f \leq u_y, \quad f = 1, \dots, \mathcal{F}, \quad (1.11e)$$

$$r_f, x_f, y_f \in \mathbb{R}, \quad f = 1, \dots, \mathcal{F}; \quad z_{tf} \in \{0, 1\}, \quad t = 1, \dots, \mathcal{T}, \quad f = 1, \dots, \mathcal{F}. \quad (1.11f)$$

The model has binary variables  $z_{tf}$  for assigning time point  $t$  to fixation  $f$ , and continuous variables  $x_f$  and  $y_f$  that indicate the center of fixation  $f$ . Bounds for  $x_f$  and  $y_f$  are constructed using  $l_x = \min_t x^t$ ,  $u_x = \max_t x^t$ ,  $l_y = \min_t y^t$ , and  $u_y = \max_t y^t$ , and further we set the values of  $\mathcal{M}_x = \max\{|x^t - l_x|, |u_x - x^t|\}$  and  $\mathcal{M}_y = \max\{|y^t - l_y|, |u_y - y^t|\}$ . Constraints (1.11c)–(1.11d) are box constraints to ensure that, if time point  $t$  is assigned to fixation  $f$  (i.e.,  $z_{tf} = 1$ ), then it lies geometrically within the appropriate square with side length  $r_f$ . Again, constraints (1.11b) represent the fundamental constraints that simply ensure, respectively, no time point is assigned to more than one fixation, every fixation contains a minimum number of points, and every fixation is composed of consecutive time points. Variable definitions and bounds are given in (1.11e)–(1.11f), while objective (1.11a) minimizes the total square fixation area, while the  $\alpha$  term accounts for the tradeoff on the number of points included.

### MISOCP Formulation: Minimize Circle Area of Fixations

Similar to the MIP formulation for minimizing the square areas of fixations in the original manuscript, the MISOCP [27] formulation attempts to balance minimizing the fixation area with inclusion of gaze points. The model contains second order conic constraints, and appears as follows.

$$\text{minimize } \sum_{f=1}^{\mathcal{F}} \left[ r_f + \alpha \sum_{t=1}^{\mathcal{T}} (1 - z_{tf}) \right], \quad (1.12a)$$

$$\text{subject to } (1.5b), (1.5c), (1.5d), \quad (1.12b)$$

$$(x^t - x_f)^2 + (y^t - y_f)^2 \leq r_f^2 + M(1 - z_{tf}), \quad t = 1, \dots, \mathcal{T}; \quad f = 1, \dots, \mathcal{F}, \quad (1.12c)$$

$$\begin{aligned} r_f &\in \mathbb{R}; \quad x_f \in [l_x, u_x], \quad y_f \in [l_y, u_y], \quad z_{tf} \in \{0, 1\}, \\ t &= 1, \dots, \mathcal{T}, \quad f = 1, \dots, \mathcal{F}. \end{aligned} \quad (1.12d)$$

Similar to the construction of MIP formulation, MISOCP formulation has continuous variables  $x_f$  and  $y_f$ , the circle center of fixation  $f$ .  $l_x, l_y, u_x$  and  $u_y$  are the bounds for  $x_f$  and  $y_f$ , but we further set the value of  $M = [(u_x - l_x)^2 + (u_y - l_y)^2]/4$ . Constraint (1.12c) is a second-order conic constraint to ensure that, if time point  $t$  is assigned to fixation  $f$  (i.e.,  $z_{tf} = 1$ ), then it lies geometrically within the appropriate circle with radius  $r_f$ . Variable definitions and bounds are given in (1.12d), and objective (1.12a) minimizes the total circumscribed fixation area.

### 1.3.4 Algorithm to Identify Densest Fixations

We provide an algorithmic approach, which we call the *fixation inner-density* (FID) filter, to identify the densest fixations from a sequence  $\mathcal{S}$  of gaze points using one of optimization formulations (1.10a)–(1.10f), (1.11a)–(1.11f) or (1.12a)–(1.12d).

---

#### Algorithm 1 Identify Densest Fixations

---

**Input:** Sequence  $\mathcal{S}$  separated into distinct chunks of consecutive  $(x, y, t)$  data  $\mathcal{C}^k$ ,  $k = 1, \dots, \mathcal{K}$ ; parameter  $\alpha$ .

- 1: Set  $\mathcal{L} \leftarrow \emptyset$ .
- 2: **for**  $k = 1, \dots, \mathcal{K}$  **do**
- 3:   Set  $\mathcal{T} \leftarrow \mathcal{T}^k$ .
- 4:   **for**  $\mathcal{F} = \mathcal{F}_{min}^k, \dots, \mathcal{F}_{max}^k$  **do**
- 5:     With  $\alpha$ , formulate and solve optimization formulation (1.10a)–(1.10f), (1.11a)–(1.11f) or (1.12a)–(1.12d).
- 6:     **if** optimal solution found **then**
- 7:       Add solution to  $\mathcal{L}$ .
- 8: **return**  $\mathcal{L}$ .

---

Algorithm 1 processes all  $(x, y, t)$  gaze-data chunks  $\mathcal{C}^k$ ,  $k = 1, \dots, \mathcal{K}$  from sequence  $\mathcal{S}$  into constituent fixations by optimizing for density using formulation (1.10a)–(1.10f), (1.11a)–(1.11f) or (1.12a)–(1.12d). Initially  $\mathcal{L}$  is empty, and by sequentially iterating over each

chunk  $\mathcal{C}^k$ ,  $k = 1, \dots, \mathcal{K}$ , it sets  $\mathcal{T}$  to the total number of gaze points  $\mathcal{T}^k$  in chunk  $\mathcal{C}^k$ , and then formulates an optimization problem for every level of  $\mathcal{F}$ . For each chunk  $\mathcal{C}^k$ , fixations of maximum density (with respect to  $\alpha$ ) are recorded and stored in  $\mathcal{L}$ .

## 1.4 Computational Experiments

We now proceed to discuss the computational performance of using Algorithm 1 to sequentially call formulations (1.10a)–(1.10f), (1.11a)–(1.11f) and (1.12a)–(1.12d), on real gaze datasets from two tasks that differ with respect to cognitive effort: online shopping [28], and solving math problems [10]. The shopping task requires participants to purchase three items in a simulated grocery store environment, while the math task requires participants to answer a set of Graduate Record Examination Math Section questions. The task of reading and processing Math GRE questions by nature requires a higher level of information processing than the shopping task, hence it is more cognitively complex.

### 1.4.1 Datasets and Equipment

We considered two datasets, one containing eye movement data from the shopping task and one from the math task. Each dataset contains  $\mathcal{R} = 10$  eye tracking recordings (indexed by  $\ell$ ). Participants were recruited from the student population in a Northeastern university of the United States. The first (shopping task) dataset was recorded by a Tobii Pro X2-30 eye tracker [29], with a frequency of 30 Hz. Each recording is between seven and twelve minutes in duration. The second (math task) dataset was recorded by a Tobii Pro TX300 [29]. Each recording is approximately five minutes in duration. This dataset was originally recorded at 300 Hz. To compare the fixation patterns between shopping and math tasks, we also downsampled this dataset for each recording by retaining the first gaze point, and every tenth point thereafter, thereby generating a new reading dataset at 30 Hz. All experiments were run on an Intel core i7-4700MQ computer with 2.40GHz and 8.0 GB RAM running 64-bit Windows 8. We used the Gurobi Optimizer [30] with Python 2.7 interface for the optimization modeling, algorithmic design, and solution process, and note that we explicitly pursue global optima for each optimization problem by using default values for the Gurobi MIPGap (1e-4) and MIPGapAbs (1e-10) parameters. MATLAB was used for designing the I-DT filter [31], while Tobii Studio was used for the I-VT filter [17]. A time limit of 12 hours (wall-clock) was present for all computational experiments.

## 1.4.2 Data Preprocessing

For each recording  $\ell$ , gaze data is preprocessed, as discussed in Section 1.3.2, by separating the data sequence  $\mathcal{S}_\ell$  into chunks  $C_\ell^k$ ,  $k = 1, \dots, \mathcal{K}_\ell$ , via saccadic events. We used the Tobii Studio I-VT filter [8, 17] to do so, together with a constant velocity threshold of  $V = 30^\circ/s$ , which is suitable for a variety of types of data under different sampling frequencies and noises [17]. Because the I-VT processing can result in one or more consecutive, non-saccadic gaze points with total duration below a theoretical minimum fixation duration which we take to be  $100ms$ ; see, e.g., [9, 13, 32], we also preemptively removed these from consideration.

Stimuli	Frequency (Hertz)	Avg # of All Points in Sequence	Avg # of Data Chunks	Avg # of Valid Data Chunks	Avg # of Points in All Data Chunks	Avg # of Points in Valid Data Chunks
Shopping Data	30	18,207	3,017	1,178	10,153	7,737
GRE Math Reading Data	30	9,058	752	575	8,092	6,822
GRE Math Reading Data	300	90,580	3,612	721	80,956	66,677

Table 1.1: Summary results on separated data with I-VT filter, averaged over ten recordings per dataset.

The minimum number of gaze points for a fixation is dependent on the frequency  $h$  (in Hertz) of the eye tracking device. From the literature, fixation durations are typically estimated in the range of  $60 - 400ms$ ; in general a minimum duration  $d_m = 100ms$  is a reasonable lower-bound for information processing to occur [9, 13, 32]. A straightforward choice of  $\mathcal{N}$  is then  $\mathcal{N} = \left\lceil \frac{h \cdot d_m}{1,000ms} \right\rceil$ . Using this we set the minimum number of gaze points to be  $\mathcal{N} = 3$  and  $\mathcal{N} = 30$  for the 30 Hz and 300 Hz datasets, respectively. Table 1.1 details summary results on the processed sequences prior to, and after, removing these small sets of points; we term as *valid* those chunks (and points) that remain after removal. In general, lower values of  $\mathcal{N}$  result in smaller, more numerous data chunks for a given data sequence. After preprocessing each  $\mathcal{S}_\ell$ ,  $\ell = 1, \dots, \mathcal{R}$ , into chunks  $C_\ell^k$ ,  $k = 1, \dots, \mathcal{K}_\ell$ , we then run Algorithm 1 using one of formulations (1.10a)–(1.10f), (1.11a)–(1.11f) or (1.12a)–(1.12d). We set  $\mathcal{F}_{min}^k = \mathcal{F}_{max}^k = 1$  for all of our experiments, as manual inspection predominantly indicated each data chunk contained a single fixation.

## 1.4.3 Evaluation Metrics

We use several metrics to evaluate the performance of our methods for each dataset and level of  $\alpha$ . The average *fixation duration*  $\delta^{avg}$  of a sequence  $\mathcal{S}$  measures, in seconds, the time spent in fixations, averaged over all fixations. The *cover rate*  $\gamma$  of a data sequence  $\mathcal{S}$  measures the ratio of the number of gaze points included in fixations, to the total number of gaze points (fixation and non-fixation) in a given data instance; [9] also reports this

measurement (“the percentage of points-of-regard that are included in fixations”). Cumulative computational runtimes are also recorded, in seconds of wall-clock time, for both the Gurobi Optimizer and Algorithm 1. Each of the metrics we consider are averaged over all ten recordings for each respective dataset.

Figure 1.3 is a small illustrative example depicting the duration and cover rate. It depicts a small data chunk (outer bounding region) containing eight gaze points obtained from 30 Hz data. The inner fixation, namely gaze points 2 through 7, has a duration of  $\delta = \frac{6}{30} = 0.2$  seconds. Supposing that the length of this recording was 8 total points, the cover rate is  $\gamma = \frac{6}{8} = 0.75$ , because six of the eight points were included in the fixation.

We also consider three distinct representations of density. The paper of [25] advocates for *minimizing* the average intra-fixation sum of distances, a measure that is inversely proportional to density (so, effectively, the optimization *maximizes density*). Hence, to keep with this convention we present our results from this perspective – the three expressions we use to characterize density are such that *smaller* magnitudes represent *greater* density.

The first of these metrics ( $\rho_1$ ) is the average pairwise distances between points within a fixation. Suppose that  $\mathcal{P}$  is the number of points contained in the fixation,  $\mathcal{P} > 2$ , and  $d_{pq}$  is the Euclidean distance between fixation points  $p$  and  $q$ . Then  $\rho_1$  is expressed as:

$$\rho_1 = \frac{\sum_{p=1}^{\mathcal{P}-1} \sum_{q=p+1}^{\mathcal{P}} d_{pq}}{\binom{\mathcal{P}}{2}}. \quad (\rho_1)$$

The second metric  $\rho_2$  is similar to  $\rho_1$ . It has the same numerator of summing the pairwise distances of all included fixation points, though the denominator is simply  $\mathcal{P}$ , which has the effect of increasing the density for fixations with greater number of points:

$$\rho_2 = \frac{\sum_{p=1}^{\mathcal{P}-1} \sum_{q=p+1}^{\mathcal{P}} d_{pq}}{\mathcal{P}}. \quad (\rho_2)$$

We consider  $\rho_2$  because of its clear relationship to objective function (1.10a) when  $\alpha = 0$ . It is meaningful to see how this metric varies under differing values of  $\alpha$ .

The third metric ( $\rho_3$ ) is the minimal area square bounding box surrounding the fixation

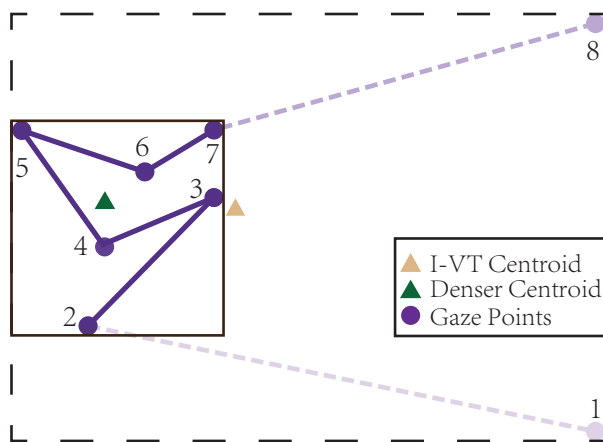


Figure 1.3: Duration  $\delta$  and cover rate  $\gamma$  for a single chunk. Our results refine those of I-VT, including only six of the eight points, yielding a denser fixation. In addition to duration and cover rate differences, the centroid shifting is also apparent.

30 Hz Shopping Data										30 Hz GRE Math Reading Data																
$\alpha$	Duration	Density Measures			Cover Rate	Center Shift	Avg Runtime (s)		$\delta^{avg}$ (s)	$\rho_1^{avg}$	$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall	$\delta^{avg}$ (s)	$\rho_1^{avg}$	$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall		
	$\delta^{avg}$ (s)	$\rho_1$	$\rho_2$	$\rho_3$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall																		
0	0.1000	21.3300	21.3300	528.3056	0.1877	6.9322	39.2	55.6	0.1000	5.2815	5.2815	94.6009	0.2637	2.7312	131.0	151.3										
3	0.1005	21.3200	21.3669	528.2375	0.1888	6.9244	129.6	146.4	0.1901	5.8623	10.9157	95.6769	0.5052	2.2796	-	-										
6	0.1075	21.3647	22.3935	527.9422	0.2040	6.8027	1,796.2	1,812.9	0.2496	6.6952	18.2435	98.9420	0.6674	1.4921	-	-										
9	0.1287	21.8642	27.2219	530.6561	0.2493	6.3368	1,911.2	1,928.1	0.2658	7.1121	21.7909	101.8035	0.7096	1.0694	-	-										
12	0.1500	22.6585	33.8933	537.9728	0.2946	5.6161	356.6	373.4	0.2722	7.3488	23.7653	103.7606	0.7258	0.8422	784.6	804.7										
15	0.1655	23.4458	40.1121	547.7675	0.3268	4.8396	216.8	233.8	0.2747	7.4884	24.7490	105.3714	0.7319	0.7191	507.9	528.0										
18	0.1752	24.1153	44.9136	557.6114	0.3466	4.1185	178.3	195.2	0.2759	7.5881	25.3391	106.7105	0.7345	0.6306	375.2	395.3										
21	0.1821	24.7263	48.9410	569.2495	0.3606	3.4843	183.0	199.9	0.2765	7.6401	25.7045	107.6167	0.7359	0.5857	219.6	239.6										
24	0.1871	25.2430	52.2651	580.8938	0.3704	2.9871	164.8	181.6	0.2768	7.7087	25.9153	109.1708	0.7365	0.5390	24.8	44.8										
27	0.1906	25.6653	54.9576	591.0722	0.3774	2.5722	136.9	153.8	0.2770	7.7516	26.0471	110.1686	0.7369	0.5068	16.5	36.6										
30	0.1934	26.0326	57.3728	601.0251	0.3831	2.1891	100.5	117.6	0.2772	7.7877	26.1815	111.2469	0.7372	0.4865	12.7	32.9										

Table 1.2: Results of Algorithm 1 & formulation (1.10a)–(1.10f) on 30 Hz shopping and GRE Math reading datasets.

divided by the number of fixation points it contains:

$$\rho_3 = \frac{(2\hat{r})^2}{\mathcal{P}}. \quad (\rho_3)$$

The minimal square side length  $2\hat{r}$  is derived from the optimal  $\hat{r}$  value in optimization formulation (1.11a)–(1.11f). A final metric, the *center shift*  $\lambda^{avg}$ , is reported in more detail in Section 1.4.4, in particular with respect to the performance of our approaches versus the standard I-VT filter.

### 1.4.4 Computational Results and Discussion

We now discuss the results of our computational experiments for our proposed methods. Table 1.2 highlights computational results from running formulation (1.10a)–(1.10f) on 30 Hz Shopping data (left) and 30 Hz GRE Math reading data (right). Table 1.3 documents the same information as Table 1.2, but using formulation (1.11a)–(1.11f). Table 1.4 details the performance of formulation (1.11a)–(1.11f) on the larger 300 Hz dataset (formulations (1.10a)–(1.10f) and (1.12a)–(1.12d) were not competitive at this higher frequency). Table 1.5 reports the computational results for formulation (1.12a)–(1.12d) on 30 Hz Shopping data (left) and 30 Hz GRE Math reading data (right). Each table has a similar format, with the rows indexed by trade-off parameter  $\alpha$ , and the columns indicating various properties discussed in Section 1.4.3, which are obtained post-optimization by averaging over all chunks in each of the ten data recordings.

The parameter  $\alpha$  represents the trade-off in emphasis between the spatial compactness versus the number of gaze points contained in a given fixation. At one extreme, a level of  $\alpha = 0$  gives no incentive for inclusion, so very compact fixations tend to form with minimal gaze points, that is, near the level of  $\mathcal{N}$ . At the other extreme, larger  $\alpha$  penalties incentivize many gaze points to be included in the fixation, likely at the expense of spatial proximity. Tension exists in between these two extremes for gaze points that, while within

30 Hz Shopping Data										30 Hz GRE Math Reading Data									
$\alpha$	Duration		Density Measures			Cover Rate	Center Shift	Avg Runtime (s)				$\delta^{avg}$ (s)	Density Measures			Cover Rate	Center Shift	Avg Runtime (s)	
	$\delta^{avg}$ (s)	$\rho_1^{avg}$	$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall	$\rho_1^{avg}$				$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall	
0	0.1006	21.5012	21.6441	522.6540	0.1888	6.9010	8.2	44.2			0.1015	5.3435	5.4356	93.7965	0.2679	2.7093	7.7	40.4	
1	0.1209	21.5645	26.7576	513.3601	0.2323	6.4350	10.3	46.6			0.2477	6.4484	19.5562	93.3753	0.6603	1.6307	6.4	39.6	
2	0.1531	22.5559	37.6929	519.0029	0.3012	5.4906	9.7	46.3			0.2669	6.9389	23.0267	89.6347	0.7120	1.1170	4.1	37.3	
3	0.1711	23.5078	45.9917	529.5563	0.3389	4.5600	8.7	45.3			0.2719	7.1749	24.1726	91.7787	0.7249	0.8920	3.5	36.6	
4	0.1812	24.2505	51.3489	541.7839	0.3595	3.8215	7.7	44.3			0.2740	7.3090	24.7753	93.4555	0.7299	0.7780	3.2	36.5	
5	0.1873	24.8422	54.9765	553.5813	0.3714	3.2194	6.9	43.6			0.2752	7.4159	25.2025	95.1061	0.7327	0.7007	3.1	36.3	
6	0.1908	25.2697	57.2888	562.2031	0.3785	2.7894	6.2	42.9			0.2759	7.4883	25.4862	96.3742	0.7344	0.6406	3.0	36.3	
7	0.1932	25.6052	59.1002	571.0291	0.3834	2.4927	5.8	42.5			0.2763	7.5522	25.7068	97.5203	0.7354	0.5861	3.0	36.5	
8	0.1951	25.9119	60.5458	580.0977	0.3871	2.1942	5.4	42.0			0.2765	7.5892	25.8133	98.1695	0.7359	0.5581	2.9	36.2	
9	0.1965	26.1606	61.7261	588.6679	0.3897	1.9573	5.1	41.8			0.2769	7.6568	26.0204	99.6355	0.7366	0.5136	2.8	36.2	
10	0.1977	26.3796	62.7787	597.2041	0.3919	1.7403	4.9	41.7			0.2770	7.6795	26.1019	100.1463	0.7368	0.4983	2.8	36.2	

Table 1.3: Results of Algorithm 1 & formulation (1.11a)–(1.11f) on 30 Hz shopping and GRE Math reading datasets.

300 Hz GRE Math Reading Data								
$\alpha$	Duration	Density Measures			Cover Rate	Center Shift	Avg Runtime (s)	
	$\delta^{avg}$ (s)	$\rho_1^{avg}$	$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall
0	0.1062	5.8589	90.1959	31.9361	0.2598	1.8150	574.3	659.5
0.1	0.2607	6.5335	241.3585	28.8872	0.6528	0.9478	364.5	454.1
0.2	0.2762	6.7828	268.4264	28.5850	0.6911	0.6739	264.7	354.7
0.3	0.2803	6.8764	277.5209	28.2034	0.7004	0.5727	207.2	299.7
0.4	0.2827	6.9654	283.6307	27.5299	0.7053	0.5046	154.7	246.6
0.5	0.2840	7.0202	287.1474	27.7181	0.7083	0.4589	119.0	212.0
0.6	0.2848	7.0571	289.3265	27.8777	0.7100	0.4300	87.0	178.1
0.7	0.2853	7.0816	290.6830	28.0161	0.7112	0.4095	67.1	159.0
0.8	0.2857	7.1100	292.1223	28.1589	0.7121	0.3880	53.9	145.1
0.9	0.2860	7.1251	292.7735	28.2548	0.7126	0.3777	43.4	136.5
1.0	0.2863	7.1483	294.0966	28.3347	0.7134	0.3612	37.7	128.8

Table 1.4: Results of Algorithm 1 & formulation (1.11a)–(1.11f) on 300 Hz GRE Math reading dataset.

30 Hz Shopping Data										30 Hz GRE Math Reading Data									
$\alpha$	Duration		Density Measures			Cover Rate	Center Shift	Avg Runtime (s)				$\delta^{avg}$ (s)	Density Measures			Cover Rate	Center Shift	Avg Runtime (s)	
	$\delta^{avg}$ (s)	$\rho_1^{avg}$	$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall	$\rho_1^{avg}$				$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	Gurobi	Overall	
0	0.1001	21.7751	21.7965	532.1699	0.1879	6.2534	16.3	80.3			0.1003	5.4527	5.4657	94.9431	0.2646	2.4653	17.2	62.4	
1	0.1170	21.9019	25.8494	529.9510	0.2243	6.1039	17.6	81.0			0.2298	6.4971	18.0447	97.7410	0.6109	1.8494	12.5	58.3	
2	0.1437	22.7085	34.9120	534.9214	0.2818	5.6520	15.6	79.4			0.2441	7.0206	20.9778	100.7641	0.6498	1.5250	9.5	56.9	
3	0.1586	23.5800	41.9778	545.8175	0.3135	5.1162	14.2	78.5			0.2471	7.2600	21.8616	103.1921	0.6572	1.4012	8.3	55.7	
4	0.1664	24.2382	46.3210	556.7186	0.3296	4.6829	13.0	77.0			0.2482	7.4053	22.2978	104.9492	0.6599	1.3425	7.7	54.9	
5	0.1711	24.7528	49.1512	567.0128	0.3390	4.3568	11.7	75.5			0.2488	7.5115	22.6194	106.4665	0.6612	1.2967	7.5	53.8	
6	0.1737	25.1590	51.0543	576.6504	0.3442	4.1230	10.9	74.9			0.2490	7.5596	22.7724	107.1763	0.6618	1.2783	7.3	53.8	
7	0.1754	25.4661	52.4020	585.1760	0.3476	3.9430	10.3	74.5			0.2492	7.6125	22.9142	108.3814	0.6621	1.2590	7.2	53.3	
8	0.1766	25.7537	53.5489	594.1787	0.3501	3.7863	9.8	74.2			0.2492	7.6496	22.9996	109.2310	0.6623	1.2387	6.9	52.1	
9	0.1776	26.0125	54.5135	602.4873	0.3518	3.6435	9.4	74.4			0.2493	7.6811	23.0626	110.1126	0.6624	1.2265	6.9	52.2	
10	0.1781	26.1922	55.1608	608.5803	0.3530	3.5430	9.3	74.1			0.2493	7.6936	23.1089	110.5536	0.6625	1.2225	6.9	51.8	

Table 1.5: Results Algorithm 1 & with formulation (1.12a)–(1.12d) on 30 Hz shopping and GRE Math reading datasets.

a given data chunk  $\mathcal{C}^k$ , are not near the center of a fixation (see, e.g., the sixth gaze point in Figure 1.2a). Due to the intrinsic and distinct interpretations of density in (1.10a) versus (1.11a) (1.12a), differing levels of  $\alpha$  are required to induce similar outcomes. For this reason we varied the range of  $\alpha$  values in Tables 1.2, 1.3, 1.5, and 1.4. Due to the higher frequency of the 300 Hz dataset, greater sensitivity with  $\alpha$  was necessary (in the

form of smaller values) to influence the results of Table 1.4.

## Runtime Discussions

For each sequence  $\mathcal{S}_\ell$ , the runtime consists of solving an optimization problem for each valid chunk  $\mathcal{C}_\ell^k$ ,  $k = 1, \dots, \mathcal{K}_\ell$ . As can be seen in Table 1.1, on average this implies solving upwards of several hundreds, and sometimes thousands, of small yet still NP-hard optimization problems. Moreover, for every computational test, there is a roughly “constant” time for processing the same dataset. This can be seen in the difference in runtimes between the “Gurobi” and “Overall” columns, with “Overall” being fairly static. Thus, the differences in runtime are largely due to the contribution of Gurobi, which experiences varying levels of computational difficulty as  $\alpha$  fluctuates. Moreover, Table 1.2, Table 1.3 and Table 1.5 exhibit the general trend that when  $\alpha$  increases, the Gurobi runtime initially increases, and then decreases. This is apparent for both the 30 Hz shopping and GRE Math reading datasets, and for both optimization formulations. This behavior is likely induced by  $\alpha$ : when  $\alpha$  is rather small yet nonzero, there is relatively greater difficulty in balancing the trade-off term in the objective of including a point or adding the penalty.

Looking across Tables 1.2, 1.3 and 1.5, in general formulation (1.10a)–(1.10f) exhibits a slower runtime performance than (1.11a)–(1.11f) and (1.12a)–(1.12d). When comparing the algorithmic performances of formulation (1.10a)–(1.10f) on shopping and GRE Math reading stimuli as reported in Table 1.2, we observe that the latter dataset exhibited much longer runtimes for several initial levels of  $\alpha$ . Generally speaking, many of the GRE Math reading data chunks were much larger than those from the shopping data. These larger data chunks, as well as the numerous new variables and constraints required to linearize formulation (1.10a)–(1.10f), are likely the reason that it returned no fixations for several levels of  $\alpha$  where the proximity-duration trade-off was most difficult to balance.

Formulation (1.11a)–(1.11f) experienced no such performance degradation on the 30 Hz datasets detailed in Table 1.3. Even so, when comparing the runtimes for the 30 Hz and 300 Hz GRE Math reading data in Tables 1.3 and 1.4, formulation (1.11a)–(1.11f) exhibits slower performance on the 300 Hz instances. It can be seen from Table 1.1 that the 300 Hz instances have larger average chunk sizes. Hence, the longer processing times are likely due to Gurobi formulating and solving (1.11a)–(1.11f) on larger data chunks. These runtime results from Table 1.4, while larger than those from Table 1.3, remain quite promising for future fixation detection on similar datasets, and for those of longer duration and at higher frequencies.

The Gurobi optimization runtime reported in Table 1.5 for formulation (1.12a)–(1.12d) is approximately twice as long as with formulation (1.11a)–(1.11f), indicating that this



formulation is more challenging for the solver.

### Fixation Duration and Cover Rate Discussions

Fixation duration  $\delta$  is a commonly-used metric in eye tracking research representing the temporal length of a fixation. For each dataset and formulation, we report in Tables 1.2, 1.3, and 1.4 the fixation duration averaged over all chunks and recordings,  $\delta^{avg}$ . When  $\alpha = 0$ , there is no incentive to include gaze points beyond the minimum necessary. Hence, the value of  $\delta^{avg}$  approaches the minimum defined length of a fixation represented by  $\mathcal{N}$ . As  $\alpha$  increases, the value of  $\delta^{avg}$  also increases, indicating that on average, fixations are containing more gaze points. Moreover, independent of dataset and formulation,  $\delta^{avg}$  experiences the greatest increase for relatively low values of  $\alpha$ .

The cover rate  $\gamma$  is a measurement that describes the ratio of points included in fixations to the total points in a data recording. For each dataset and formulation, we report the cover rate averaged over all recordings,  $\gamma^{avg}$ . As  $\alpha$  increases,  $\gamma^{avg}$  exhibits an increasing trend in Tables 1.2, 1.3, 1.4 and 1.5. Independent of the formulation, the largest  $\gamma$  increases occur at slightly different values of  $\alpha$ . For the GRE Math reading data, the largest jump in  $\gamma$  occurs immediately after  $\alpha$  transitions from 0 to the first nonzero value. For the shopping data, however, the greatest  $\gamma$  increase occurs somewhat subsequent to the initial nonzero  $\alpha$  transition. After these larger jumps,  $\gamma$  increases at a decreasing rate.

### Density Metric Discussions

The three density metrics discussed in Section 1.4.3 are averaged over all chunks in each of the ten data recordings, and reported in Tables 1.2, 1.3, 1.4 and 1.5. Recall that, in keeping with [25], density is largest for small  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  values. Some general trends across all experiments is that  $\rho_1^{avg}$  never exceeds  $\rho_2^{avg}$ . This is a relatively straightforward observation because, while  $\rho_1^{avg}$  and  $\rho_2^{avg}$  have identical numerators,  $\rho_1^{avg}$  always has at least as large of a denominator (and often larger). The  $\rho_3^{avg}$  metric evaluates the ratio of the minimal bounding box area to the number of points in the fixation, hence is a slightly different metric and often differs in magnitude from  $\rho_1^{avg}$  and  $\rho_2^{avg}$ .

For all datasets and formulations, the general trend is for  $\rho_1^{avg}$ ,  $\rho_2^{avg}$ , and  $\rho_3^{avg}$  to increase as  $\alpha$  increases, implying that, on average, fixations decrease in density. For all three metrics, both the numerator and denominator will increase as  $\alpha$  increases, hence there are some slight fluctuations as  $\alpha$  varies, and among the three metrics,  $\rho_3^{avg}$  exhibits the greatest variation for early values of  $\alpha$ . Another observation is that the difference between  $\rho_1^{avg}$  and  $\rho_2^{avg}$  increases as the value of  $\alpha$  increases. This increase is largely attributed to the difference in denominators of  $\rho_1^{avg}$  and  $\rho_2^{avg}$ . For the 300 Hz dataset, as can be seen in Table 1.4, there is a much larger difference between  $\rho_1^{avg}$  and  $\rho_2^{avg}$ . This is again due

to the linear versus quadratic nature of the denominators; with the 300 Hz dataset, the value of the minimum duration threshold  $\mathcal{N}$  is much larger, implying that each fixation should contain many more points.

Another important observation is that, independent of formulation, the fixations in GRE Math reading data both exhibit greater density than those for the shopping data, as well as feature longer durations. As it is known that longer fixations are representative of higher levels of information processing [1], the results in our study give further support that the math task was cognitively more demanding than the shopping task. Moreover, our results also provide evidence that fixations for more cognitively complex tasks are denser than less demanding tasks. This in turn is a valuable insight for studies that use eye tracking to capture information processing behavior at the physiological level.

### Comparing Our Methods with Existing Methods

Having already observed that fixation duration is strongly influenced by the level of  $\alpha$ , which controls for inner-density, we now demonstrate that our approaches can fine-tune the locational precision of the I-VT method. We introduce the *center shift*  $\lambda^{avg}$ , which measures the straight-line (Euclidean) distance, in pixels, between the I-VT fixation centroid and the densest fixation centroid, averaged over all fixations. These values are reported in Tables 1.2, 1.3, 1.4 and 1.5. It can be clearly seen that lower  $\alpha$  values yield larger  $\lambda^{avg}$  values than do higher  $\alpha$  values. This is because smaller  $\alpha$  values increase the inner-density of the resulting fixations, and in so doing, the fixation centroids become more centralized due to the exclusion of some peripheral points existing in data chunks.

#### 1.4.5 Polynomial-time Algorithm to Identify Single Fixation per Chunk

When considering finding one fixation in each data chunk, we further devised a polynomial-time algorithm to find the densest fixation. The algorithm examines all possible partitions by identifying two critical gaze points: the one that begins, and the one that ends the fixation (recall, by definition, that fixations must contain gaze points that are consecutive in time).

#### Polynomial-time Algorithm to Identify Densest Fixation

For each begin-end pair, the algorithm computes the corresponding objective function, and when a smaller objective function value is found, updates the optimal begin-end pair and associated objective function value. The cost of this algorithm is  $\mathcal{O}(KT^2)$ .

---

**Algorithm 2** Polynomial-time Algorithm for Densest Fixations Identification

---

**Input:** Sequence  $\mathcal{S}$  separated into distinct chunks of consecutive  $(x, y, t)$  data  $\mathcal{C}^k$ ,  $k = 1, \dots, \mathcal{K}$ ; parameter  $\alpha$ ; number of fixation to be clustered  $\mathcal{F} = 1$ .

- 1: Set  $\mathcal{L} \leftarrow \emptyset$ .
- 2: **for**  $k = 1, \dots, \mathcal{K}$  **do**
- 3:   Set  $d \leftarrow \infty$ .
- 4:   Set  $l \leftarrow \emptyset$ .
- 5:   Set  $\mathcal{T} \leftarrow \mathcal{T}^k$ .
- 6:   **for**  $i = 1, \dots, \mathcal{T} - 1$  **do**
- 7:     **for**  $j = i + \mathcal{N}, \dots, \mathcal{T}$  **do**
- 8:       Set  $z_{t1} = 1, t = i, \dots, j$ , then calculate (1.10a), (1.11a) or (1.12a) with  $\alpha$ . Let the result be  $d'$ .
- 9:       **if**  $d > d'$  **then**
- 10:           $d \leftarrow d', l \leftarrow (i, j)$ .
- 11:   Add solution  $l$  to  $\mathcal{L}$ .
- 12: **return**  $\mathcal{L}$ .

---

### Computational Experiments for Polynomial-time Algorithm

We perform Algorithm 2 on the Math reading dataset as the experiments. The objective function is chosen as (1.11a). Since the values for evaluation metrics are the same as Tables 1.2, 1.3, 1.5, and 1.4, we only report the runtime of Algorithm 2 for performance comparison.

Algorithm	Frequency(Hz)	$\alpha$ for 30Hz/300Hz Math Reading Data										
		0	1/0.1	2/0.2	3/0.3	4/0.4	5/0.5	6/0.6	7/0.7	8/0.8	9/0.9	10/1.0
Polynomial-time Algorithm	30	1.74	1.76	1.74	1.72	1.74	1.73	1.72	1.73	1.74	1.73	1.71
MILP for Minimizing Square Areas	30	7.7	6.4	4.1	3.5	3.2	3.1	3.0	3.0	2.9	2.8	2.8
Polynomial-time Algorithm	300	255.0	255.4	254.2	254.6	256.8	257.2	255.7	255.9	257.0	256.0	248.5
MILP for Minimizing Square Areas	300	574.3	364.5	264.7	207.2	154.7	119.0	87.0	67.1	53.9	43.4	37.7

Table 1.6: Runtime comparison with the Gurobi optimization runtime on formulation (1.11a)–(1.11f) for minimizing the square areas for fixations for 30Hz and 300Hz math reading dataset.

The runtime of Algorithm 2 for the two datasets are fairly static, without the influence of varying levels of  $\alpha$ . For low frequency dataset, it outperforms formulation (1.11a)–(1.11f) significantly, however, with the increase of  $\alpha$ , the gap becomes smaller and when  $\alpha = 10$ , the difference is reduced to 1.1 seconds, which is a fairly acceptable result. The more interesting finding is that for the high frequency dataset, when  $\alpha$  is greater than 0.3, the MIP formulation is faster than the polynomial-time algorithm. When comparing to the runtime in polynomial-time algorithm, we can clearly see the effect of balancing the trade-off term in the objective of including a point or adding the penalty when  $\alpha$  is rather small. Our MIP formulation is more competitive than this basic polynomial-time algorithm on high frequency dataset.

## 1.5 Conclusions

Our proposed fixation inner-density (FID) filter both builds on strengths of both the I-VT and I-DT filters, and avoids shortcomings. Velocity-based methods serve as a suitable method to group a gaze data sequence into fixation chunks by removing saccadic points (as per the I-VT filter). Moreover, excluding consecutive gaze points for which the duration is below a realistic threshold is also a useful way to remove gaze points unrelated to fixations (similar to the I-DT filter). By optimizing for inner-density, one of the fixation micro-patterns, on each resulting data chunk, we essentially use a dispersion-based approach to identify fixations. A key difference is that, rather than a static threshold used in I-DT, our dispersion threshold is dynamic – this is directly expressed by the variable  $r$ , characterizing bounding region side length, that is minimized in formulation (1.11a)–(1.11f) and (1.12a)–(1.12d). By doing so, we minimize the inclusion of fringe points in fixations and thus improve the accuracy of fixation duration and location. Hence, our methods are a refinement of both approaches.

Figure 1.4 contrasts the performance of the raw I-VT filter with the performance of formulation (1.11a)–(1.11f) and  $\alpha = 0.1$  on the same data sequence depicted in Figure 1.1. The callouts denote saccadic points by stars, fixation points by circles, and points that are eliminated by our approach by triangles. The smallest 2D boundaries for both approaches are also drawn. Some I-VT fixations (e.g. Fixation 1) contain nearly 35% more points as compared to ours (66 vs. 49 points). This refinement can have a large affect on key gaze metrics such as fixation duration and center.

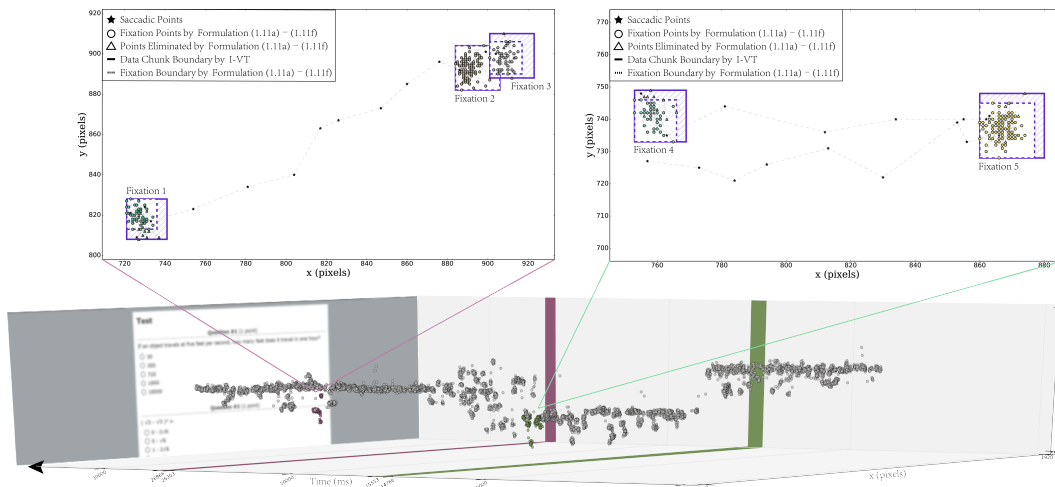


Figure 1.4: Comparing fixations identified with standard I-VT, versus formulation (1.11a)–(1.11f),  $\alpha = 0.1$ , in the gaze stream depicted in Figure 1.1. Some I-VT fixations contain nearly 35% more points than our approach.

Our computational experiments on two actual shopping and GRE Math reading datasets yielded encouraging results, in particular formulation (1.11a)–(1.11f) is quite

robust to the larger 300 Hz GRE Math reading dataset over a variety of parametrized  $\alpha$  values. The reasonable runtimes suggest further scalability for formulation (1.11a)–(1.11f). Moreover, all formulations are able to identify fixations with greater density than the standard I-VT filter, revealing that finer detail is available than what the I-VT can otherwise provide.

Our computational findings have important implications for eye tracking research. First, they show that considering fixations at a more refined scale can provide important insights into cognitive processing levels, as our computational experiments reveal that tasks with greater cognitive complexity featured longer-lasting fixations with heightened density. Hence, the results provide a rationale and theoretical direction for studying behavior via a new metric in user experience and human-computer interaction studies. Additionally, our results demonstrate that inner-density is a valuable concept; when combined with optimization-based approaches, it is a useful and novel way to identify fixations. In particular, the inner-density parameter  $\alpha$  provides a previously unavailable level of control for studying focused fixation, which we believe will prove fruitful in many fields of study that use fixation duration and location to identify behavior, including marketing, user experience, human-computer interaction, and medical diagnosis.

# Chapter 2

## Measuring Focused Attention Using Fixation Inner-Density<sup>1</sup>

In this chapter, we more thoroughly investigate our proposed fixation inner-density (FID) filter with respect to the performance of a widely used method of fixation identification, the I-VT filter. To do so we use a set of measures that investigate the distribution of gaze points at a micro-level, that is, the patterns of individual gaze points within each fixation. Our results show that in general fixations identified by the FID filter are significantly denser and more compact around their fixation center. They are also more likely to have randomly distributed gaze points within the square box that spatially bounds a fixation. Our results also show that fixation duration is significantly different between the two methods. Because fixation is a major unit of analysis in behavioral studies and fixation duration is a major representation of the in-tensity of attention, awareness, and effort, our results suggest that the FID filter is likely to increase the sensitivity of such eye tracking investigations into behavior.

### 2.1 Introduction

A fixation is the collection of gaze points that are near to one another in both time and proximity, a denser collection of gaze points within a fixation represents higher level of focused attention, and thus higher level of cognitive processing [10]. Thus, we proposes formulations (1.10a)–(1.10f), (1.11a)–(1.11f) and (1.12a)–(1.12d), which we called as Fixation Inner-density (FID) filter, to group gaze points into fixations based on their inner-density property. Identifying fixations using the FID filter naturally eliminates those gaze points that are near to tolerance settings. How gaze points are dispersed in a fixation affects

---

<sup>1</sup>W. Liu, S. Djamasbi, A. C. Trapp, “Measuring Focused Attention Using Fixation Inner Density,” *Human-Computer Interaction International Conference 2018 (HCII2018)*, Las Vegas, Nevada, July 2018.

fixation metrics such as the duration and center location, and there is evidence that the FID filter reduces the possibility of skewing these metrics.

In this chapter we translate raw gaze data into fixation using the I-VT and FID filters. We demonstrate that fixations processed by the FID filter are superior in terms of three key fixation *micro-patterns* than those that are processed by the I-VT filter. First, they are *denser*. Second, the extent to which points are dispersed within a fixation is *smaller*. Third, the points within a fixation are more likely to be uniformly distributed. This investigation is important because the compactness and the patterns of distribution of gaze points can directly affect fixation metrics, such as fixation duration and fixation center position, that are commonly used in eye tracking studies to assess viewing behavior. This study is the first to investigate such fixation *micro-patterns* or properties of the distribution of gaze points within an individual fixation.

## 2.2 Background

One popular fixation identification algorithm is the I-VT filter. It identifies fixations by gaze point velocity. If the velocity exceeds the predefined threshold  $V$ , the corresponding gaze point is identified as a saccade, otherwise it is categorized as a fixation point. I-VT filter is efficient and practical; however, it has the drawback of ignoring the information about the spatial arrangement of individual gaze points within a distinct fixation. Some fixation metrics can express the distribution of points within a fixation. One such metric is fixation inner-density, which was introduced by [10] and further re-fined in Chapter 1. Fixation inner-density represents user focus, and [10] has validated that fixation inner-density is correlated with normalized fixation duration and average pupil dilation variation during fixation. The FID filter uses optimization-based techniques to optimize for inner-density, which means that it selects a set of candidate gaze points that guarantees there is no better set with respect to the objective function of maximizing fixation inner-density. Fixation inner-density improves upon previous fixation identification methods because it combines both the temporal and the spatial aspects of the fixation into a single metric that evaluates the compactness of a fixation.

As the problem of fixation identification is a type of time-series clustering, it shares the commonality that interpreting clustering results is somewhat subjective in nature. Hence, the choice of an appropriate metric will directly affect the formation of the clusters. While density and dispersion properties can be measured in various ways, they are inherently positively related to the number of gaze points in a fixation, and negatively related to the area occupied by the constituent points. We next discuss some important metrics to evaluate density and dispersion properties within fixations.

## 2.3 Methodology

We consider two representative ways of measuring fixation inner-density, both of which are advocated in Section 1.4.3. The first density metric ( $\rho_1$ ) is the average pairwise distance between points within a fixation. The second density metric ( $\rho_3$ ) is the minimum area square bounding box surrounding the fixation divided by the number of fixation points it contains.

For both the  $\rho_1$  and  $\rho_3$  density metrics, small values imply greater density. A third metric, Standard Distance ( $SD$ ), measures the dispersion of gaze points around the fixation center.  $SD$  is a common metric in the Geographic Information System (GIS) literature, that evaluates how points are distributed around the fixation center [33]. Similar to standard deviation,  $SD$  quantifies the dispersion of a set of data values. Hence, the  $SD$  score is a summary statistic representing the compactness of point distribution. Smaller  $SD$  values correspond to gaze points that are more concentrated around the center ( $\bar{X}_f, \bar{Y}_f$ ) of fixation  $f$ , expressed as:

$$\bar{X}_f = \frac{\sum_{i=1}^{\mathcal{T}^k} x_i}{\mathcal{T}^k}, \bar{Y}_f = \frac{\sum_{i=1}^{\mathcal{T}^k} y_i}{\mathcal{T}^k}. \quad (2.1)$$

The standard distance of fixation  $f$ ,  $SD_f$ , is:

$$SD_f = \sqrt{\frac{\sum_{i=1}^{\mathcal{T}^k} (x_i - \bar{X}_f)^2}{\mathcal{T}^k} + \frac{\sum_{i=1}^{\mathcal{T}^k} (y_i - \bar{Y}_f)^2}{\mathcal{T}^k}}. \quad (2.2)$$

Spatial pattern analysis can also be examined in measuring the fixation gaze point distribution pattern. The Average Nearest Neighbor ( $ANN$ ) [33] is used to measure the degree to which fixation gaze points are clustered, versus randomly distributed, within a fixation bounding area. A fixation resulting from focused gaze toward a single area of interest would tend to exhibit a more uniformly distributed pattern, with greater  $ANN$  values. The  $ANN$  ratio is calculated as the average distance between each point and its nearest neighbor, divided by the expected average distance between points if a random pattern is assumed.  $ANN$  values greater than one imply that the fixation gaze points are dispersed; as this ratio decreases, fixation gaze points increasingly exhibit clustering.

The four metrics  $\rho_1$ ,  $\rho_3$ ,  $SD$  and  $ANN$  will be used to evaluate three aspects of inner fixation patterns: fixation inner-density, fixation points dispersion, and their distribution. We expect fixations identified with the FID filter to be denser and more uniformly distributed than those identified with the I-VT filter. Our density assertion, which stems from the method of fixation identification, helps to test whether the FID filter does indeed more accurately group individual gaze points into focused attention. Our assertion that



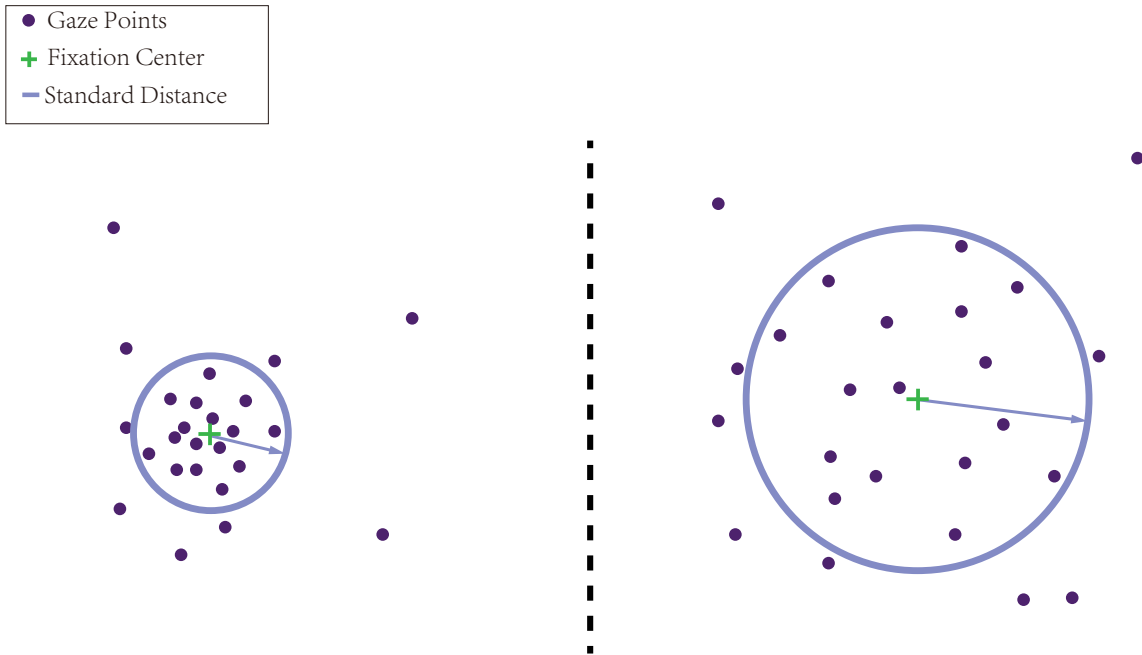


Figure 2.1: An illustrative depiction of standard distance,  $SD$ . When considering an identical number of gaze points,  $SD$  is smaller when points are more compactly distributed around the center (left); when they are more dispersed,  $SD$  becomes larger (right).

gaze points identified with the FID filter are more randomly distributed stems from the argument that if a fixation is compact, that is it has high inner-density, it is more likely to have a more uniform distribution around its center.

In addition to the above assertions, we also examine the impact of FID and I-VT filters on fixation duration  $\delta$  and center location  $\lambda^{avg}$ .

## 2.4 Experimental Evaluation

We begin this section by describing the specific context of our eye tracking datasets and experiments. We then compare the I-VT and FID filters with the aforementioned four metrics, and discuss our findings.

### 2.4.1 Dataset and Equipment

We perform our experiments on eye movement datasets obtained from a total of 28 university students who were assigned to read a text passage shown on a standard desktop computer monitor. Prior to the experiment, each participant completed a brief eye-calibration process lasting less than one minute. We used the Tobii X300 eye tracker [29] to collect participants eye-movements. The software version is 3.2.3 and the sampling rate

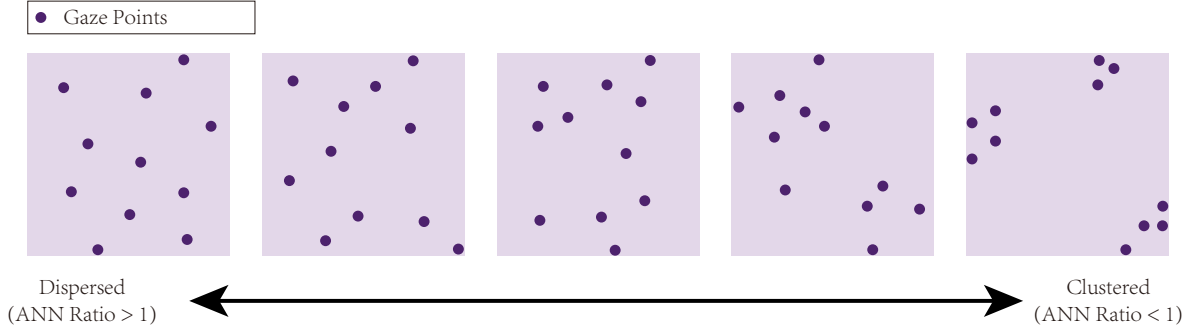


Figure 2.2: Illustrating the  $ANN$  ratio as the distribution of gaze points change within an identical minimum square bounding box.

was set to 300 Hz. The 28 recordings were further analyzed using an Intel core i7-6700MQ computer with 3.40 GHz and 16.0 GB RAM running 64-bit Windows 10. Matlab 2016a and Python 2.7 were used for additional data analysis and processing.

## 2.4.2 Data Preprocessing

For each eye tracking record, we used the Tobii Studio I-VT filter [17] to generate I-VT fixation identification results. The velocity threshold  $V$  was set to 30/s, which is the recommended threshold in [17]. The minimum fixation duration is set to 100ms which is the theoretical minimum fixation duration suggested by other eye tracking studies [9, 13].

We further used the results of the I-VT fixation identification as the input data chunks for the mixed integer programming formulation (MIP) for minimizing square area of fixations from formulation (1.11a)–(1.11f). The Gurobi Optimizer 7.5.1 [30] is used as the solver. The FID filter is parametrized by a manually assigned constant  $a$  that enables decision-makers to have fine-tuned control over the density. We varied  $a$  from 0 to 1 by steps of 0.1 on one randomly selected eye tracking record and examined the fixation identification results manually. When  $\alpha=0.1$ , the clustering result appeared the most reasonable, and averaging  $\rho_3$  values over all fixations yielded the smallest value, suggesting the algorithm finds the (averaged) densest fixations at  $\alpha=0.1$  comparing to other  $a$  levels. Therefore, we set  $\alpha=0.1$  when running the FID filter on the other 27 records. In the following evaluations, we discard the record used for selecting  $a$  to avoid data snooping.

## 2.4.3 Experimental Results

After discarding the single record above, in this section we first report our statistical analyses from the point of view of a single record. Subsequently, we expand it to all 27 of the (remaining) records in our dataset.

## 2.4.4 Comparing I-VT and FID Filters for a Single Record

Fixation inner-density and the distribution of gaze points within an individual fixation are micro-patterns in gaze data. Such patterns are relatively difficult to evaluate by averaging over all eye tracking records. To more thoroughly investigate micro-patterns, we first illustrate the comparison results on the eye tracking record of one randomly selected participant. Toward the end of this section, the comparison summary over all recordings is also included.

For this gaze data record, there are 9,788 gaze points and 110 fixations. We calculated fixation inner-density metrics  $\rho_1$  and  $\rho_3$  on each individual fixation. The resulting average of both  $\rho_1$  and  $\rho_3$  from the I-VT filter is larger than that of FID, which indicates that fixations from the FID filter are denser than those in I-VT filter result. We performed a paired  $t$ -test with the following hypothesis:

$$H_0 : \bar{\rho}_{I-VT} = \bar{\rho}_{FID},$$

$$H_a : \bar{\rho}_{I-VT} > \bar{\rho}_{FID}.$$

The  $t$ -test on both  $\rho_1$  and  $\rho_3$  returns a p-value smaller than 0.05, so at a 95% confidence level we reject  $H_0$ , which implies  $\bar{\rho}_{I-VT}$  is statistically larger than  $\bar{\rho}_{FID}$ .

Fixation Density	I-VT		FID ( $\alpha = 0.1$ )		$t$ -test	
	Mean (pixel)	STD (pixel)	Mean (pixel)	STD (pixel)	p-value	Result
$\rho_1$	6.769	2.382	5.994	1.961	<0.0001	Reject
$\rho_3$	7.690	10.450	5.112	3.920	0.0025	Reject

Table 2.1: Comparison of fixation density for I-VT and FID filters.

The  $SD$  metric measures the dispersion of fixation points around their center. Table 2.2 reveals that the  $SD$  mean and standard deviation for the I-VT filter are larger than that of the FID filter. We also performed a paired  $t$ -test when comparing the  $SD$  metric. The hypotheses are:

$$H_0 : \overline{SD}_{I-VT} = \overline{SD}_{FID},$$

$$H_a : \overline{SD}_{I-VT} > \overline{SD}_{FID}.$$

With the same 95% confidence level as the previous test, the  $t$ -test result rejects the  $H_0$ . It indicates that the FID filter tends to identify fixations having points that are more dispersed around the center. It further demonstrates that identifying fixations by

optimizing for fixation inner-density yields fixations with more compact regions.

	I-VT		FID ( $\alpha = 0.1$ )		<i>t</i> -test	
	Mean (pixel)	STD (pixel)	Mean (pixel)	STD (pixel)	p-value	Result
SD	5.5033	2.189	4.746	1.616	<0.0001	Reject

Table 2.2: Comparison of *SD* for I-VT and FID filters.

Finally, we perform a hypothesis test using the *ANN* ratio [33] to see if the gaze points are randomly distributed in a fixation region:

$H_0$  : gaze points are randomly distributed within fixation region,

$H_a$  : gaze points are not randomly distributed within fixation region.

If the hypothesis test results in a small p-value, we would reject the  $H_0$  because of the small probability that the fixation gaze points are randomly distributed in their fixation region. The *ANN* hypothesis test is rather sensitive with respect to the bounding region used to cover all fixation points in an individual fixation. Therefore, we perform two experimental results using  $A_{sq}$  and  $A_{rt}$ , respectively, to represent fixation area. Table 2.3 reports the count of fixations (out of 110) for which  $H_0$  is rejected at 95% confidence level, implying that there is statistical evidence that fixation points are not randomly distributed. Table 2.3 reveals that, under both fixation regions, more fixations appear to not be randomly distributed when using the I-VT filter. Moreover, the difference between the I-VT and FID filters is greater under the  $A_{sq}$  region. This may be due to  $A_{sq}$  typically being larger than  $A_{rt}$ , as the FID filter specifically minimizes the square area of fixations.

	I-VT		FID ( $\alpha = 0.1$ )	
	$A_{sq}$	$A_{rt}$	$A_{sq}$	$A_{rt}$
# of Fixations Rejecting $H_0$	95	60	61	50

Table 2.3: Comparison of *ANN* for I-VT and FID filters, reporting the count of fixations (out of 110) for which  $H_0$  is rejected.

We now compare fixation duration  $\delta$  and fixation center for the I-VT and FID filters. Fixation duration ( $\delta$ ) is a commonly used metric in eye tracking research. We compare the average fixation duration on I-VT and FID filters with the hypotheses that

$$H_0 : \overline{FD}_{I-VT} = \overline{FD}_{FID},$$

$$H_a : \overline{FD}_{I-VT} > \overline{FD}_{FID}.$$

The paired  $t$ -test result shows that  $\overline{FD}_{FID}$  is significantly smaller than  $\overline{FD}_{I-VT}$  at a 95% confidence level. This outcome may be due to the FID filter eliminating fixation points and refining the fixation region of each of the fixation chunks from the I-VT filter.

	I-VT		FID ( $\alpha = 0.1$ )		t-test	
	Mean (second)	STD (second)	Mean (second)	STD (second)	p-value	Result
Fixation Duration	0.250	0.151	0.204	0.167	<0.0001	Reject

Table 2.4: Comparison of fixation duration for I-VT and FID filters.

Fixation center is also a basic feature to represent fixation location, used in the depiction the scan path of eye movement. We introduce the center shift( $\lambda^{avg}$ ), which is the Euclidean distance between the fixation center of the I-VT filter and that of the FID filter. The 110 fixations within the eye tracking record generates mean and standard deviation (STD) of the center shift data as reported in Table 2.5.

	Mean (pixel)	STD (pixel)
Center Shift	0.881	1.617

Table 2.5: Statistics of fixation center shift between I-VT and FID filter.

When examining the mean and STD of center shift, it may be inferred that the difference of fixation center is negligible. The bivariate distribution of center shift depicted in Figure 3 displays the long tail distribution in both x and y axis. The 90% quantile of  $x$ ,  $y$  is 0.922 and 1.308 respectively. It shows that while the refined results of the FID filter can skew some I-VT fixation centers, most of the time the center shift remains in a fairly small range.

### 2.4.5 Comparing I-VT and FID Filters for All 27 Remaining Records

The results reported above were for a single eye tracking record. The average number of gaze points for all remaining 27 records is 10,959, and the average number of fixations is 127.7. Table 2.6 reports the results of the corresponding hypothesis tests for  $\rho_1$ ,  $\rho_3$ ,  $SD$  and fixation duration on all the 27 eye tracking records. We find that zero record does not reject the corresponding  $H_0$  in the  $t$ -test for  $\rho_1$ ,  $SD$  and fixation duration, and two for  $\rho_3$ . This analysis shows that the FID filter finds denser and more compact fixations than I-VT filter holds for most of eye tracking records in our dataset in terms of for  $\rho_1$ ,

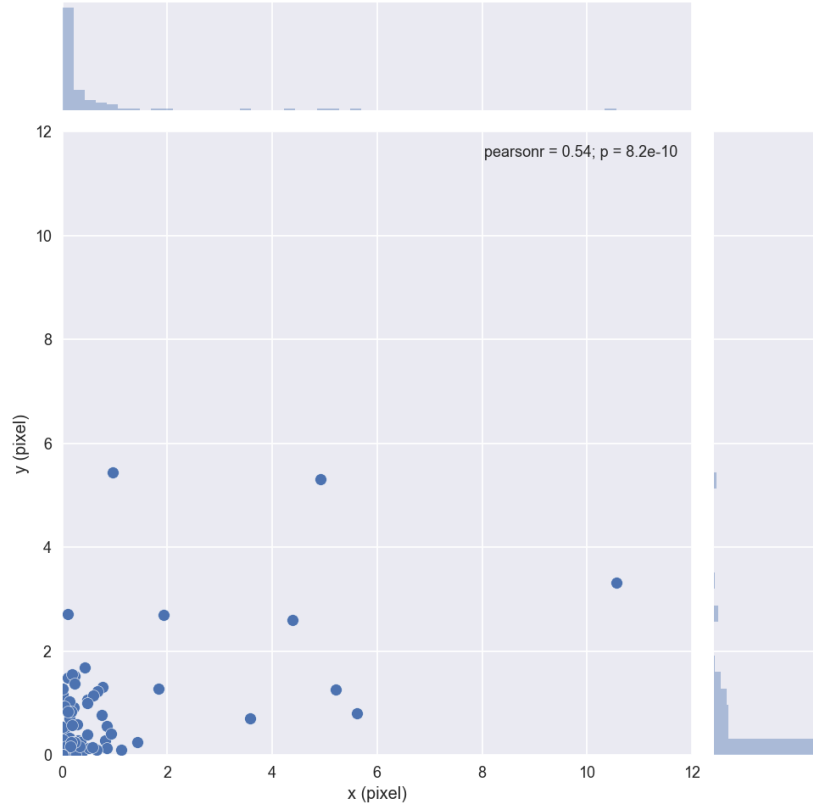


Figure 2.3: The bivariate distribution of center shift in  $x$ ,  $y$  coordinates.

$\rho_3$  and  $SD$ .

	$\rho_1$	$\rho_3$	$SD$	Fixation Duration
# of Records That Do Not Reject $H_0$	0	2	0	0

Table 2.6: Summary of hypothesis test results for 27 eye tracking records.

	$\rho_1$	$\rho_3$	$SD$
# of Records That Do Not Reject $H_0$	0	2	0

Table 2.7: Summary of hypothesis test results for 27 eye tracking records.

We calculate the center shift between all I-VT and FID filter fixation pairs; the bivariate distribution result is shown in Figure 2.4. The distribution on either  $x$  or  $y$  direction is again a long tail distribution. The 90% quantile value of  $x$ ,  $y$  is 2.095 and 2.411 respectively. Figure 2.4 shows only a few points that are far away from the origin, indicating that the FID filter identification results can indeed change the fixation center location, though this occurred relatively infrequently in our dataset.

We also run the  $ANN$  hypothesis test on each recording and calculate the count of

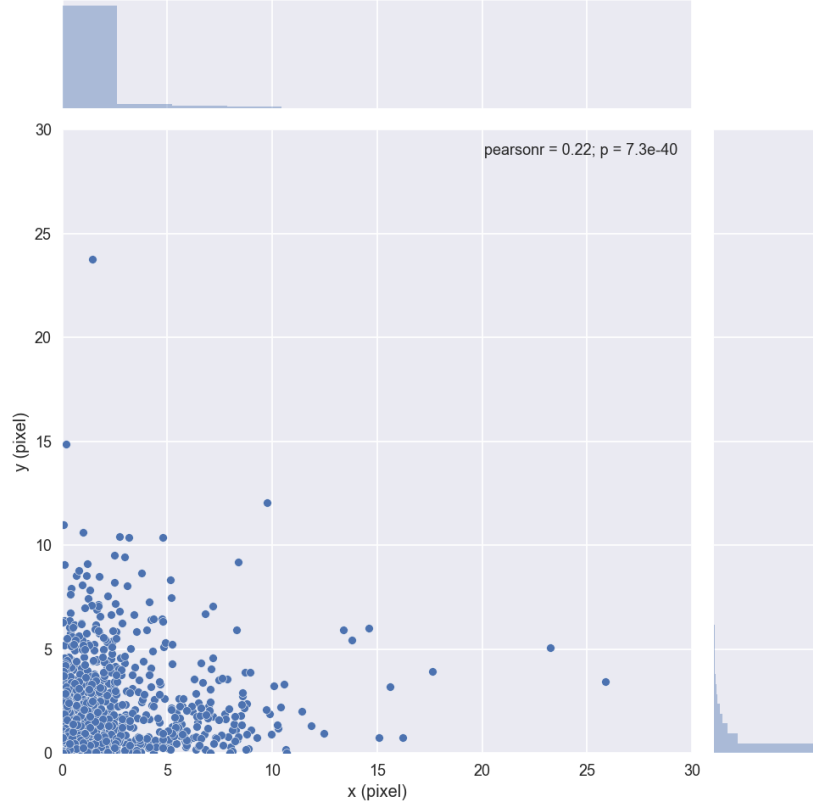


Figure 2.4: The bivariate distribution of center shift for all fixations.

fixations ( $FC$ ) for which the  $ANN$  hypothesis test  $H_0$  ( $FC - ANN$ ) is rejected over all recordings. The average is reported in Table 2.8. Both the mean and the standard deviation resulting from the FID filter are smaller than that of the I-VT filter. We compare the  $FC$  results from the I-VT and FID filters by the paired  $t$ -test with 95% confidence level and the following hypotheses:

$$\begin{aligned}
 H_0 &: \overline{FC}_{I-VT} = \overline{FC}_{FID}, \\
 H_a &: \overline{FC}_{I-VT} > \overline{FC}_{FID}.
 \end{aligned}$$

The first row in Table 2.8 shows that when bounding the fixation region by  $A_{sq}$ ,  $\overline{FC}_{FID}$  is significantly smaller than  $\overline{FC}_{I-VT}$ . It indicates the general trend that the inner gaze points of fixations resulting from the FID filter tend to be randomly distributed. As for  $A_{rt}$ , the  $t$ -test result also reject  $H_0$ , implying that the same conclusion could be drawn on  $A_{rt}$ .

Fixation Region	I-VT		FID ( $\alpha = 0.1$ )		$t$ -test	
	Mean (count)	STD (count)	Mean (count)	STD (count)	p-value	Result
$A_{sq}$	109.1	40.0	70.2	27.4	<0.0001	Reject
$A_{rt}$	74.5	30.1	64.2	24.6	0.0002	Reject

Table 2.8: Comparison of  $FC - ANN$  for I-VT and FID filters over all recordings.

## 2.5 Conclusions

Our results show that the FID filter, as compared to I-VT filter, does indeed identify fixations that are denser and more compact around the center, and more uniformly distributed patterns found in fixation bounding regions. These properties have major implications for two important fixation metrics that are widely used in eye tracking analysis: Fixation duration and location. Our results show that the two filters tend to result in significantly different fixation durations. The results displayed in Figure 2.3 and Figure 2.4 provide evidence that in some cases FID filter can result in quite different fixation centers comparing to I-VT filter. It is important to note that the data used in our study was gathered when users were reading an online text passage, which typically generates more focused fixations. Future investigation using different stimuli are needed to extend the generalizability of these results and to see whether the micro-level differences, including fixation duration and center location, observed in this study between FID and I-VT filters change for different tasks (e.g., reading more challenging text passages, viewing a picture, or browsing a website). For example, in this study we used a reading task which typically results in compact fixations. Using a browsing task may result in much larger differences in fixation center location, because gaze points within fixations in browsing tasks tend to more dispersed. The metrics introduced in this study to compare fixations at a micro level serve to refine the analysis of eye movements to a deeper level. Future studies, however, are needed to validate and extend our findings.

The results of this study contribute in two ways to eye tracking studies that examine user behavior. First, they show that researchers can identify focused attention with the FID filter and thereby improve the sensitivity of their analysis with regard to duration and center location of intense attention. Second, the micro-analysis introduced in this study provides a new way to compare gaze points within a fixation. This is important because it allows researchers to examine relationships between eye movements and behavior at a much smaller unit of analysis, namely fixation micro-patterns.



# Chapter 3

## Outlier-Aware, Density-Based Gaze Fixation Identification

Of great interest in eye-tracking studies and behavioral research are *fixation* events, which are indicative of attention and awareness. While fixations enable downstream interpretation of gaze phenomena, and empower decision making, eye-tracking *imprecision*, such as what arises from system noise, calibration errors or erratic eye movements, can lead to *outlier* points. To resolve such inaccuracies, we propose FID<sup>+</sup>: outlier-aware fixation identification via fixation inner-density. This work extends the FID filter in Chapter 1. We represent this problem through a novel mixed-integer optimization formulation, and subsequently strengthen the formulation using two geometric arguments to provide enhanced bounds. We show that neither bound dominates the other, and that both are effective in reducing the overall solution runtime. Our experiments on real gaze recordings demonstrate that accommodating for the reality of fixation outliers enhances the ability to identify fixations with greater density.

### 3.1 Introduction

The purpose of fixation identification is to recognize distinct eye movement events in raw gaze data. Primary methods for identifying fixations are those based on velocity, and those based on gaze dispersion. Velocity-based methods group consecutive gaze points into fixations by using the fact that fixation points have lower velocities than saccade points. The I-VT filter is the classic velocity-based method [8]. Dispersion-based methods use the assumption that fixation points usually lie closer to each other than saccade points. The I-DT filter is a well-known dispersion-based method [8]. While these methods serve as baseline implementations for separating fixations and saccades, their abilities to identify fixations have limited precision, thus skewing fixation properties [7, 34] that hinder

downstream research relying on these foundational properties.

To overcome these limitations, Chapter 1 introduces the notion of fixation *inner-density*. Inner-density, as a representation of fixation micro-patterns, incorporates both the temporal and spatial aspects of the fixation. When combined, these aspects reveal significant and previously undiscovered information about attention. Inner-density addresses limitations of existing methods, such as lack of sensitivity to peripheral fixation points, as well as the misrepresentation of fixation properties. Chapter 1 used integer optimization techniques to identify fixations in a sequence of gaze points by optimizing for inner-density, also known as the FID filter. In particular, the formulation (1.11a)–(1.11f) presented minimizes the square area of a bounding box around the constituent fixation points. Minimizing the square area, which can be observed in Figure 3.1, is equivalent to minimizing the apothem  $r$  (half of the side length, as introduced in Chapter 1) due to monotonicity. Computational results demonstrated that this optimization-based approach is efficient and effective in identifying denser fixations than the current I-VT method. Though promising, one limitation of the FID filter is handling noise within gaze data, as well as erratic eye movements within fixations. The optimization model in Chapter 1 enforces that within a single fixation, all fixation points must be temporally adjacent. However, in reality there at times exists occasional noise within fixation events. Hence, it is worthwhile to allow for some small deviations in the sequence, for example if a stray gaze point exists between two larger clusters of gaze points in the same region. In this case, it may be preferable to omit this gaze point. Doing so requires an adaptation of the mathematical formulation in Chapter 1 to allow for *outliers* between successive sequences of fixation points.

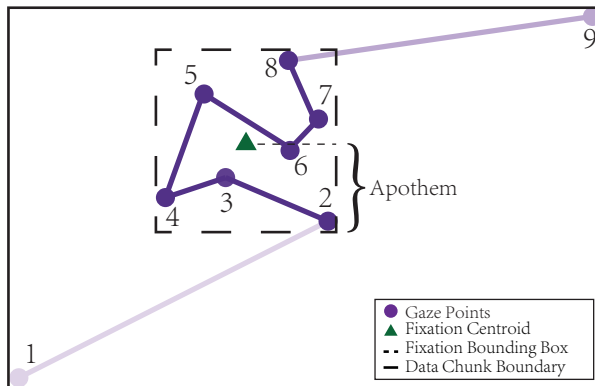


Figure 3.1: Illustration of fixation and its apothem (side half-length) identified in a gaze data chunk; FID filter: minimizing apothem of fixation bounding box.

This chapter augments the FID filter by allowing noise points to be eliminated within the fixation. We propose an enhanced mathematical optimization formulation (FID<sup>+</sup>) to account for this *outlier sensitivity*. To the best of our knowledge, this work and Chapter 1 are the first, and only, approaches to identify fixations in gaze data by optimizing for density. The addition of a new set of budget-constrained binary variables accounts for the condition of where a gaze point is labeled as an outlier. In conjunction with the existing binary variables that indicate whether a gaze point is labeled as a fixation point,

we introduce two new constraint sets that together represent time consistency in light of outlier gaze points. While the new formulation accurately remedies the aforementioned limitation, it does so at the cost of additional complexity. Thus, we present two algorithmic techniques to tighten lower bounds on the size of the apothem (which is minimized) to improve the computational performance.

The remainder of this chapter is organized in the following manner. In Section 3.2 we provide background on data quality and fixation outliers. In Section 3.3 we present FID<sup>+</sup>, a novel mixed integer programming (MIP) formulation for detecting fixations with outlier sensitivity. We subsequently provide two geometric arguments to strengthen the optimization formulation by enhancing the lower bounds on the apothem of the bounding box, and demonstrate that both are advantageous (we show that neither technique dominates the other). Section 3.4 details the computational experiments on real eye-tracking data, including a discussion on its observed performance. Finally, we conclude the chapter and discuss future work in Section 3.5.

## 3.2 Background

High-quality gaze data is the foundation of generating valid and reproducible behavioral research results. As illustrated in Figure 3.2, *Accuracy* and *precision* are the two highlighted aspects measured for eye-tracking data quality. The reference location, denoted with a “+”, is where the participant is asked to fixate. *Accuracy*, also called offset, refers to the shift between the recorded gaze position location, and the actual reference location. *Precision* refers to the variance of the recorded positions to the reference location [35–37].

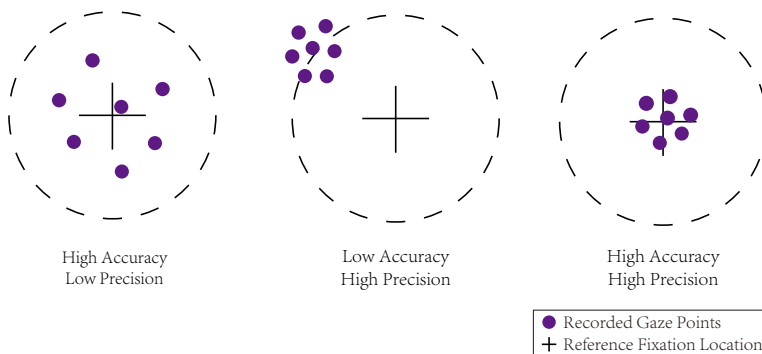
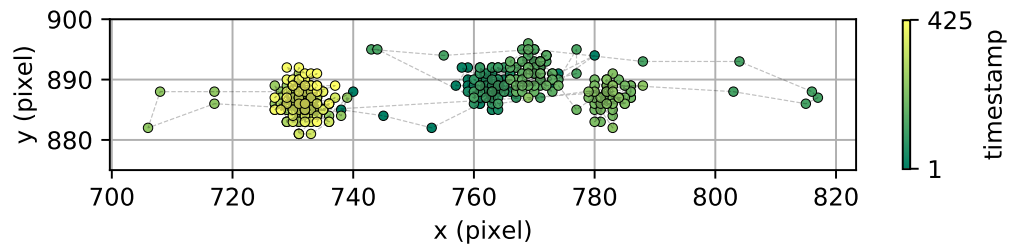


Figure 3.2: Illustration of accuracy and precision for measuring gaze data quality. Accuracy is the difference between the centroid of grouped recorded gaze points, and an actual reference fixation location. Precision is the variance of the gaze point dispersion in a fixation.

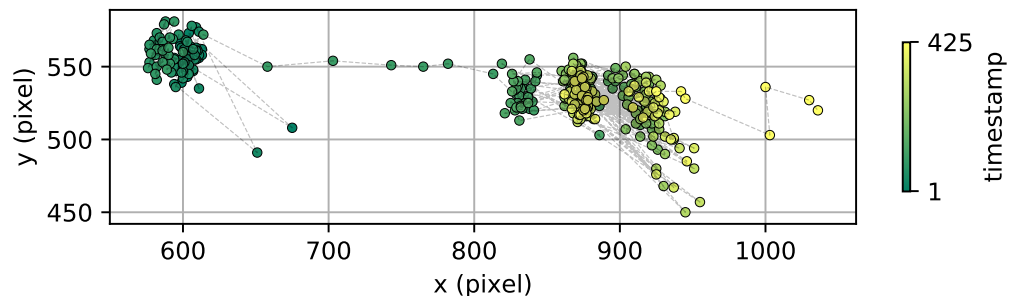
Inaccuracy and imprecision can be attributed to multiple factors: eye-tracking cameras [36], algorithms for capturing eye movements [36], experimental design [38], system

issues (such as sensor noise, data loss) [39], and various participant characteristics (such as glasses, astigmatism, eye color, head movements) [36]. Poor data precision leads to noisy gaze samples, which can challenge the reliability of fixation identification algorithms.

Figure 3.3(a) illustrates a raw gaze sequence with 425 points collected by a Tobii Pro TX300 [29] eye-tracking device, while Figure 3.3(b) shows a noisy raw gaze sequence with the same length also from the same device. Gaze points in Figure 3.3(a) show explicit clusters at the location of fixations. However, the clusters in Figure 3.3(b) contain multiple stray points, and those points appear to drift to the same direction from their temporally adjacent points. The fixation patterns in Figure 3.3(b) will inevitably contain some noise points in a long fixation gaze point sequence. Such noise points should be viewed as *Fixation Outliers*, and subsequently be eliminated from fixations.



(a) Well-calibrated gaze data in two dimensions recorded by eye-tracking device.

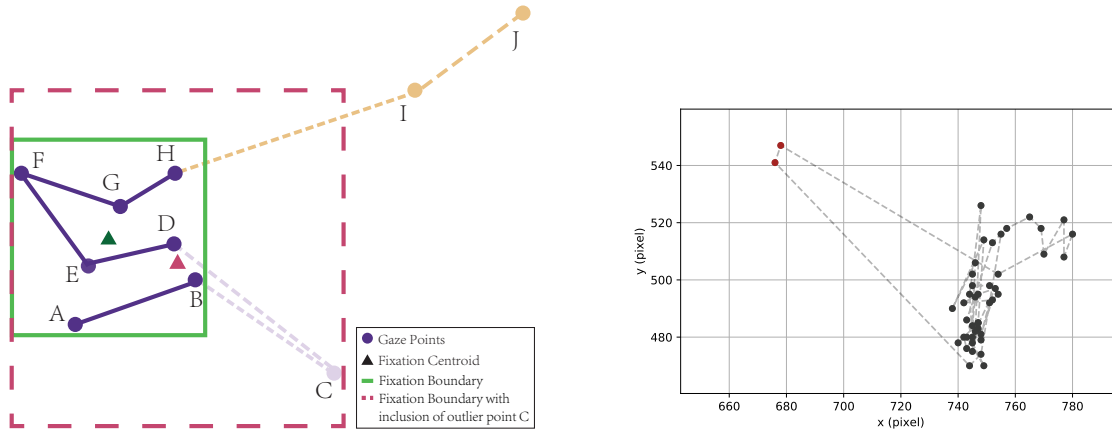


(b) Noisy raw gaze data in two dimensions recorded by eye-tracking device.

Figure 3.3: Comparison between normal gaze data and noisy data.

Fixation outliers can have substantial effects on the precision of fixation metrics, such as the number, and duration, of fixations [35]. Also impacted is *dwelling time*, a commonly used measurement of gaze duration in eye-tracking research for entering and remaining in an area of interest [12]. As illustrated in Figure 3.4(a), when the point C is included as a fixation point, the square fixation bounding region increases significantly and the fixation centroid shifts away from its original position. Figure 3.4(b) shows an actual example of possible fixation outliers appearing in real gaze data.

The FID filter described in Chapter 1 is unable to account for fixation outliers because the strict nature of the constraint set (1.3) that every fixation contains only consecutive



(a) When excluding fixation outlier point C, we get a tighter, denser fixation, with better metrics: the center shifts from the red triangle to the green; the density increases as the bounding region decreases.

(b) Example in raw gaze data: the intermediate red gaze points (top left) are far from the main cluster of gaze points, indicating the potential to be fixation outliers.

Figure 3.4: Influence of fixation outliers on fixation metrics.

gaze points in time. However, Figure 3.4 highlights the benefits of eliminating fixation outliers. Therefore, to enable the FID filter to account for outlier sensitivity, we extend the approach in Chapter 1.

### 3.3 Mathematical Developments

From a gaze sequence  $\mathcal{S}$  with  $\mathcal{T}$  points  $(x^t, y^t)$ ,  $t = 1, \dots, \mathcal{T}$ , we seek to identify fixation points to constitute  $\mathcal{F}$  fixations. The fixation identification problem discussed in Chapter 1 requires each fixation to contain at least  $\mathcal{N}$  points for information processing to occur, and those points must be temporally adjacent. Define  $\mathcal{TF}$  binary variables  $z$ , with  $z_{tf} = 1$  if gaze point  $t$  is included in fixation  $f$ , and 0 otherwise. Of the three formulations presented in Chapter 1 for FID filter in finding dense fixations, we focus on *Minimize Square Area of Fixations* (formulation (1.11a)–(1.11f)). The formulation bounds each fixation with a two-dimensional square box of minimal area; it achieves a minimum area by equivalently minimizing the apothem of the square,  $r_f$ . The model incorporates a non-negative parameter  $\alpha$  into the objective function that balances the trade-off between the inclusion of additional gaze points and the compactness of the fixation region.

#### 3.3.1 Decomposition Principle

The gaze sequence length  $\mathcal{T}$  can easily reach the hundreds of thousands gaze points, and the number of fixations can likewise be in the thousands. Formulation (1.11a)–(1.11f)

is valid for any number of gaze points  $\mathcal{T}$  and fixations  $\mathcal{F}$ . This includes subsequences obtained after applying the decomposition principle discussed in Chapter 1. This process separates a gaze data sequence into distinct data chunks  $\mathcal{C}^k$ ,  $k = 1, \dots, \mathcal{K}$ , with data chunk separated by one or more saccade points as identified by benchmark filters such as the I-VT filter. After the decomposition, minimal fixations remain within each data chunk, and formulation (1.11a)–(1.11f) can identify  $\alpha$ -densest fixations efficiently in each chunk. Again, we term this approach the *FID filter*. We also apply this decomposition principle in the  $\text{FID}^+$  filter.

### 3.3.2 $\text{FID}^+$ Filter: Detecting Fixation Outliers in Gaze Data

In this section we present the insights for extending the mathematical formulation to identify fixations with outlier sensitivity.

#### New Variables for Outlier Detection

We extend formulation (1.11a)–(1.11f) to additionally classify a small portion of gaze points within the identified fixations as *fixation outliers*. Although they lie in the interior of a fixation time sequence, they are not identified as fixation points (i.e.,  $z_{tf} = 1$ ). Define  $\mathcal{TF}$  binary variables  $q$ , with  $q_{tf} = 1$  if gaze point  $t$  is an outlier in fixation  $f$ , and 0 otherwise.

#### Fixation Outlier Budget

We propose a budget  $\mathcal{P}$  to allow some small number of outlier points. One reasonable value for  $\mathcal{P}$  is a percent  $p$  of the total number of gaze points  $\mathcal{T}$  in the chunk, so that  $\mathcal{P} = \lceil p\mathcal{T} \rceil$ . Hence, the sum of outlier points over all fixations should be less or equal to  $\mathcal{P}$ :

$$\sum_{f=1}^{\mathcal{F}} \sum_{t=1}^{\mathcal{T}} q_{tf} \leq \mathcal{P}. \quad (3.1)$$

Alternatively,  $\mathcal{P}$  can be set to any user-defined, positive integer.

#### Relaxation from Absolute Time Consistency

Constraint set (1.3) in Chapter 1 ensures the included points within each fixation must be consecutive in time. Fixation  $f$  terminates once a consecutive time pair  $(z_{tf}, z_{t+1,f})$  appears as  $(1,0)$  among all the possible values  $\{(0,0), (0,1), (1,1), (1,0)\}$ . When  $(z_{tf}, z_{t+1,f})$  equals to  $(1,0)$ , the right-hand side becomes zero, ensuring that  $z_{jf} = 0$ , for all  $j : t + 1 \leq j \leq \mathcal{T}$ . It guarantees that the remainder of the points in the chunk are not

included in this fixation. For the other possible values of  $(z_{tf}, z_{t+1,f})$ , the right-hand side is either  $(\mathcal{T} - t)$  or  $2(\mathcal{T} - t)$ , so the constraint set becomes vacuous. Thus, for a fixation  $f$ , a starting gaze point at time  $a$  and an ending point at time  $b$ , constraint set (1.3) ensures  $z_{tf}$  is assigned in the following fashion: i)  $z_{tf} = 0$ , ii)  $z_{tf} = 1$ .

However, when a set of outlier gaze points  $\mathcal{E} \subset \{a + 1, \dots, b - 1\}$  appears between the starting and ending fixation points, as indicated by  $q_{tf} = 1$ , the corresponding  $z_{tf}$  should be assigned to zero. The assignment ii) changes to  $z_{tf} = 0$  and  $z_{tf} = 1$ .

The consecutive pair  $(z_{tf}, z_{t+1,f})$  equals to  $(1,0)$  not only happens at the termination of  $f$ , but can also occur when point  $t+1$  is identified as an outlier, i.e.,  $q_{t+1,f} = 1$ . When fixation  $f$  terminates,  $(z_{tf}, z_{t+1,f})$  is  $(1,0)$  and  $q_{t+1,f}$  should be assigned as zero. Following this interpretation, we extend the constraint set from (1.3) to (3.2) by relaxing the assumption that fixation points must be consecutive in time:

$$\sum_{j=t+1}^{\mathcal{T}} z_{jf} \leq (\mathcal{T} - t)(1 - z_{tf} + z_{t+1,f} + q_{t+1,f}), \quad t = 1, \dots, \mathcal{T} - 1; \quad f = 1, \dots, \mathcal{F}. \quad (3.2)$$

When  $q_{t+1,f} = 0$ , indicating point  $t + 1$  is not an outlier for fixation  $f$ , the right-hand side in (3.2) equals zero when consecutive time pair  $(z_{tf}, z_{t+1,f})$  equals  $(1,0)$ . Thereby it ensures the following variable  $z_{jf}$ , for all  $j : t + 1 \leq j \leq \mathcal{T}$  must be zero, which means fixation  $f$  terminates as it may no longer include any gaze points. When  $q_{t+1,f} = 0$ , the constraint set has the same impact as constraint set (1.3). However when  $q_{t+1,f} = 1$ , the constraint set induces no restrictions under any alternatives of  $(z_{tf}, z_{t+1,f})$ , because the right-hand side is always at least  $(\mathcal{T} - t)$ . Thus, the consecutive variables  $z_{jf}$ , for all  $j : t + 1 \leq j \leq \mathcal{T}$  may still be assigned to one. Therefore, the subsequent gaze points from  $t + 1$  to  $\mathcal{T}$  can be included in fixation  $f$  and the assignment of  $(1,0)$  to the pair  $(z_{tf}, z_{t+1,f})$  no longer delineates the end of the fixation.

## Controlling the Position of Outliers

While constraint set (3.2) generalizes the condition of strict time consistency, there is no implication on the values that points  $z_{jf}$ , for all  $j : t + 1 \leq j \leq \mathcal{T}$  can take when  $q_{t+1,f} = 1$ . In the absence of any other constraints, this may cause a fixation to be decomposed into multiple components. To ensure that every fixation  $f$  has consecutive gaze points formed by only fixation points ( $z_{tf} = 1$ ) and outlier points ( $q_{tf} = 1$ ), the following set of constraints can be incorporated:

$$q_{tf} \leq q_{t+1,f} + z_{t+1,f}, \quad t = 1, \dots, \mathcal{T} - 1, \quad f = 1, \dots, \mathcal{F}. \quad (3.3)$$

Constraint set (3.3) ensures that if  $q_{tf} = 1$ , the next gaze point at  $t + 1$  must be classified as a fixation point ( $z_{t+1,f} = 1$ ) or a fixation outlier ( $q_{t+1,f} = 1$ ). When  $q_{tf} = 0$ , the constraint is always valid. While this constraint set technically allows both  $z_{t+1,f} = 1$  and  $q_{t+1,f} = 1$ , there are scarce outlier points available by (3.1), and so gaze points are classified as outliers only when it is beneficial for the objective, that is, when subsequent gaze points are classified as fixation points. Constraint set (3.3) introduces  $\mathcal{TF} - \mathcal{F}$  additional constraints.

### 3.3.3 Minimizing Square Area of Fixations with Outlier Sensitivity

We now present the final MIP formulation for FID<sup>+</sup>: outlier-aware fixation identification via density optimization. Note that the extensions discussed in Section 3.3.2 can also be applied to *Minimize Average Intra-Fixation Sum of Distances* (formulation (1.10a)–(1.10f)) and *Minimize Circle Area of Fixations* (formulation (1.12a)–(1.12d)).

$$\text{minimize } \sum_{f=1}^{\mathcal{F}} \left[ r_f + \alpha \sum_{t=1}^{\mathcal{T}} (1 - z_{tf}) \right], \quad (3.4a)$$

$$\text{subject to } \sum_{f=1}^{\mathcal{F}} z_{tf} \leq 1, \quad t = 1, \dots, \mathcal{T}, \quad (3.4b)$$

$$\sum_{t=1}^{\mathcal{T}} z_{tf} \geq \mathcal{N}, \quad f = 1, \dots, \mathcal{F}, \quad (3.4c)$$

$$\sum_{j=t+1}^{\mathcal{T}} z_{jf} \leq (\mathcal{T} - t)(1 - z_{tf} + z_{t+1,f} + q_{t+1,f}),$$

$$t = 1, \dots, \mathcal{T} - 1, \quad f = 1, \dots, \mathcal{F}, \quad (3.4d)$$

$$q_{tf} \leq q_{t+1,f} + z_{t+1,f}, \quad t = 1, \dots, \mathcal{T} - 1, \quad f = 1, \dots, \mathcal{F}, \quad (3.4e)$$

$$\sum_{f=1}^{\mathcal{F}} \sum_{t=1}^{\mathcal{T}} q_{tf} \leq \mathcal{P}, \quad (3.4f)$$

$$x_f - r_f - \mathcal{M}_x(1 - z_{tf}) \leq x^t \leq x_f + r_f + \mathcal{M}_x(1 - z_{tf}), \quad t = 1, \dots, \mathcal{T}, \quad (3.4g)$$

$$y_f - r_f - \mathcal{M}_y(1 - z_{tf}) \leq y^t \leq y_f + r_f + \mathcal{M}_y(1 - z_{tf}), \quad t = 1, \dots, \mathcal{T}, \quad (3.4h)$$

$$r_f \geq 0, \quad l_x \leq x_f \leq u_x; \quad l_y \leq y_f \leq u_y, \quad f = 1, \dots, \mathcal{F}, \quad (3.4i)$$

$$z_{tf} \in \{0, 1\}, \quad q_{tf} \in \{0, 1\}, \quad t = 1, \dots, \mathcal{T}, \quad f = 1, \dots, \mathcal{F}. \quad (3.4j)$$

Formulation (3.4a)–(3.4j) uses binary variables  $z_{tf}$  to assign time point  $t$  to fixation  $f$ . It incorporates binary variables  $q_{tf}$  to identify outlier points in each fixation  $f$ . Objective



function (3.4a) minimizes the sum of fixation square apothems, penalizing the number of excluded points with parameter  $\alpha$ . Constraints (3.4b) and (3.4c) are the fundamental constraints indicating that a time point can be assigned to at most one fixation, and each fixation contains at least  $\mathcal{N}$  points. Constraint set (3.4d) relaxes fixation point assignment from absolute time consistency, while constraint set (3.4e) ensures points identified as outlier points are succeeded by either outlier or fixation points. Constraint set (3.4f) ensures the number of identified outlier points is within the fixation outlier budget  $\mathcal{P}$ . Constraints (3.4g)–(3.4h) ensure that the identified points in fixation  $f$  present in the fixation bounding box with center  $(x_f, y_f)$  and apothem  $r_f$ . Variable definitions and bounds are listed in (3.4i)–(3.4j).

While formulation (3.4a)–(3.4j) is correct and detects fixation and outlier points, initial computational testing on larger instances revealed that, while strong feasible solutions were quickly found, the MIP solver Gurobi [30] experienced difficulty proving optimality.

### 3.3.4 Deriving Lower Bounds on $r_f$

Objective function (3.4a) minimizes the apothem  $r_f$  of the bounding box encompassing the fixation points. While feasible solutions to (3.4a)–(3.4j) representing strong upper bounds are quickly computed using the MIP solver Gurobi [30], the lower bounds often exhibit only gradual progress toward convergence, likely due to poor relaxation strength from constraint set (3.4d).

To accelerate the computational proof of optimality, we present geometric arguments that can strengthen lower bounds on  $r_f$ . We algorithmically preprocess the gaze point sequences to identify lower bounds  $\ell$  on  $r_f, f = 1, \dots, \mathcal{F}$ .

#### Deriving Lower Bounds on $r_f$ via Sliding Windows

Consider identifying  $\mathcal{F}$  fixations from a gaze sequence with  $\mathcal{T}$  total points, each of which requires at least  $\mathcal{N}$  fixation points to ensure cognitive processing occurs [1]. Further, suppose the entire budget of  $\mathcal{P}$  outlier points is used in a fixation with the minimum number of points  $\mathcal{N}$ . Lemma 1 states that there will be at least one subsequence separated by outlier points that contains at least  $\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor$  consecutive gaze points.

**Lemma 1** *Suppose for fixation  $f$ , the fixation point sequence  $s_f$  has length  $\mathcal{N}_f$ , and it is decoupled into subsequences by  $\mathcal{P}_f$  fixation outliers. There always exists a subsequence  $s$  of  $s_f$  with length of at least  $\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor$  points.*

**Proof.** The average length of all subsequences in fixation  $f$  is  $\frac{\mathcal{N}_f}{\mathcal{P}_f+1}$ , hence there is at least one subsequence  $s$  whose length is greater than or equal to  $\frac{\mathcal{N}_f}{\mathcal{P}_f+1}$ . Because  $\mathcal{N}_f \geq \mathcal{N}$

and  $\mathcal{P}_f \leq \mathcal{P}$  by (3.4c) and (3.4f), this implies  $\frac{\mathcal{N}_f}{\mathcal{P}_f} \geq \frac{\mathcal{N}}{\mathcal{P}} \geq \frac{\mathcal{N}}{\mathcal{P}+1} \geq \lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor$ . Thereby the length of  $s$  is also greater than or equal to  $\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor$ .  $\blacksquare$

For fixation  $f$ , the apothem  $r_f$  represents a minimum bounding box covering all included fixation points, starting from a gaze point at time  $a$  to an ending gaze point at time  $b$ . The apothem of the bounding box must satisfy  $r_f \geq \frac{1}{2} \max\{|x^i - x^j|, |y^i - y^j|\}$  for all the point pairs  $(i, j) : a \leq i < j \leq b$ . The apothem  $r_f$  of the bounding box is monotonically nondecreasing as the number of points in the range  $[a, b]$  increases. Thus, a conservative global lower bound  $\ell_1$  on  $r_f$  can be derived from the individual lower bounds originating from the distance arising from  $t$ , to  $t$  shifted by the minimum number of consecutive gaze points,  $\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor$ . By considering all pairs of points  $(t, t + \lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor - 1)$  for  $t = 1, \dots, \mathcal{T} - \lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor + 1$ , we obtain a lower bound on  $r_f$ . Finding  $\ell_1$  can be accomplished in polynomial time. For each begin-end point pair, we compute the corresponding minimum bounding length  $\ell'_1$ :

$$\ell'_1 = \frac{1}{2} \max \left\{ |x^i - x^j|, |y^i - y^j|, t \leq i < j \leq t + \left\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \right\rfloor - 1 \right\}. \quad (3.5)$$

When a smaller  $\ell'_1$  is found, we update  $\ell_1$  to be  $\ell'_1$ . The cost of this method is  $\mathcal{O}(\mathcal{T} - \lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor)$ , that is, it is linear in the number of gaze points  $\mathcal{T}$ . This method is summarized in Algorithm 3.

---

**Algorithm 3** Determine Valid Lower Bound  $\ell_1$

---

**Input:** Gaze sequence  $\mathcal{S}$  with length  $\mathcal{T}$ ; fixation outlier budget  $\mathcal{P}$ ; minimum number of fixation points  $\mathcal{N}$ .

**Output:** Lower bound  $\ell_1$  on the fixation apothem  $r_f$ .

- 1: Set  $\ell_1 \leftarrow \infty$ .
  - 2: **for**  $t = 1, \dots, \mathcal{T} - \lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor + 1$  **do**
  - 3:   Calculate the minimum bounding length  
 $\ell'_1 = \frac{1}{2} \max\{|x^i - x^j|, |y^i - y^j|, t \leq i < j \leq t + \lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor - 1\}$ .
  - 4:   **if**  $\ell'_1 < \ell_1$  **then**
  - 5:     Set  $\ell_1 \leftarrow \ell'_1$ .
  - 6: **return**  $\ell_1$ .
- 

**Theorem 1** For a gaze sequence  $\mathcal{S}$ ,  $\ell_1$  is a valid lower bound for  $r_f, f = 1, \dots, \mathcal{F}$ , i.e.  $\ell_1 \leq r_f$ .

**Proof.** Suppose there exists  $\ell_1 > r_f$  from Algorithm 3. By Lemma 1, we can find a subsequence  $s$  with length of at least  $\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor$ . We further truncate  $s$  by sequentially eliminating the points from either the beginning or the end, until the remaining sequence length is exactly  $\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor$ . The remaining sequence constitutes a new sequence  $s'$ , and let its

bounding region apothem be  $\ell_1'$ . Because  $s'$  is contained in  $s$ , it has fewer fixation points than fixation  $f$ . The lower bound on the bounding box apothem, by the construction in (3.5), is a nondecreasing function in the number of points in the fixation, thus we conclude that  $\ell_1' \leq r_f$ . Therefore,  $\ell_1' < \ell_1$ , which contradicts the fact that  $\ell_1$  is the minimal bounding box apothem for all the consecutive gaze subsequences with length of  $\lfloor \frac{\mathcal{N}}{p+1} \rfloor$ . Thus, the original statement holds. ■

### Deriving Lower Bounds on $r_f$ via Smallest Enclosing Squares

For a gaze sequence of  $\mathcal{T}$  points, the apothem length of the smallest enclosing square covering  $\mathcal{N}$  points, irrespective of temporal adjacency, is a valid lower bound  $\ell_2$  for  $r_f$ ,  $f = 1, \dots, \mathcal{F}$ . We adapt Algorithm 4 from [40] for finding the smallest square bounding box of  $\mathcal{N}$  points for each input gaze sequence. Algorithm 4 first sorts the gaze points at  $x$ -decreasing order and sweeps each point. Hence, the algorithm sweeps points from right to left. When sweeping at point  $t$ , the current  $x^t$  is recorded as  $p_1$ . From the points lying at the right-hand side of the vertical line drawn by  $p_1$ , it finds a set of points  $V$  whose  $x$ -axis value is in the range of  $[x_t, x_t + \ell_2]$ ,  $y$ -axis value is in the range of  $[y_t - \ell_2, y_t + \ell_2]$ , where  $\ell_2$  is the smallest apothem of the enclosing square identified thus far. It then finds the squares which exactly cover  $\mathcal{N}$  points and their left side is on the vertical line through  $p_1$  and bottom side is on the line through a point in  $V$ . At each  $p_1$ , the algorithm sweeps a horizontal line  $q_2$  from the top point to the bottom point of  $V$ . Two binary search trees  $A$  and  $B$  are maintained to store every point  $(x, y)$  above  $q_2$ . If the horizontal distance  $x - p_1$  is greater than the vertical distance  $y - q_2$ , the point is stored in  $A$  in increasing  $x$ -order. Otherwise it is stored in  $B$  in increasing  $y$ -order. For each  $q_2$ , the element at rank  $k$  in the set  $(A - p_1) \cup (B - q_2)$  is selected. This is the side length for a square that covers  $k$  points in the area from the top of  $V$  to  $q_2$ . We compute  $\ell_2'$  as the half of the side length, and if  $\ell_2' < \ell_2$ , we update  $\ell_2$  to be  $\ell_2'$ .

---

**Algorithm 4** Determine Valid Lower Bound  $\ell_2$ 

---

**Input:** Gaze sequence point set  $\mathcal{S}$  with length  $\mathcal{T}$ ; minimum number of gaze points  $\mathcal{N}$ .

**Output:** Lower bound  $\ell_2$  on the fixation apothem  $r_f$

- 1: Sort points in  $\mathcal{S}$  at  $x$ -decreasing order.
- 2: Set  $\ell_2 \leftarrow \infty$ .
- 3: Set  $P \leftarrow$  empty balanced binary search tree.
- 4: **for**  $t = 1, \dots, \mathcal{T}$  **do**
- 5:    $p_1 = x^t$ .
- 6:    $xMax = x^t + \ell_2$
- 7:    $yMax = y^t + \ell_2$ .
- 8:    $yMin = y^t - \ell_2$ .
- 9:   Insert a new node into  $P$ , key= $y^t$ , value= $(x^t, y^t)$ .
- 10:   Set  $V \leftarrow \emptyset$
- 11:   **for** node  $p \in P$  **do**
- 12:     Select the value of node  $p$ ,  $(x^p, y^p)$  from  $P$ .
- 13:     **if**  $x^p \leq xMax$  **then**
- 14:       **if**  $yMin \leq y^p \leq yMax$  **then**
- 15:         Add  $(x^p, y^p)$  to  $V$ .
- 16:       **else**
- 17:         Delete  $p$  from  $P$ , i.e.,  $P = P \setminus p$ .
- 18:     **if**  $|V| \geq \mathcal{N}$  **then**
- 19:       Sort points in  $V$  at  $y$ -decreasing order.
- 20:       Set  $A \leftarrow$  empty balanced binary search tree.
- 21:       Set  $B \leftarrow$  empty balanced binary search tree.
- 22:       **for**  $i = 1, \dots, |V|$  **do**
- 23:         Select  $q = V[i] = (x^q, y^q)$  from  $V$ .
- 24:         Set  $q_2 = y^q$ .
- 25:         Insert a new node into  $A$ , key= $x^q$ , value= $(x^q, y^q)$ .
- 26:         **for**  $a \in A$  **do**
- 27:         Select  $a.value$ ,  $(x^a, y^a)$  from  $A$ .
- 28:         **if**  $y^a - q_2 > x^a - p_1$  **then**
- 29:         Delete  $a$  from  $A$ , i.e.,  $A = A \setminus a$ .
- 30:         Insert a new node into  $B$ , key= $y^a$ , value= $(x^a, y^a)$ .
- 31:       **if**  $i \geq \mathcal{N}$  **then**
- 32:         Find the key  $k$  at rank  $\mathcal{N}$  in  $(A - p_1) \cup (B - q_2)$ .
- 33:          $\ell'_2 = \frac{1}{2}k$ .
- 34:         **if**  $\ell'_2 < \ell_2$  **then**
- 35:         Set  $\ell_2 \leftarrow \ell'_2$ .
- 36: **return**  $\ell_2$ .

---

**Theorem 2** For a gaze sequence  $\mathcal{S}$ ,  $\ell_2$  is a valid lower bound for  $r_f$ ,  $f = 1, \dots, \mathcal{F}$ , i.e.  $\ell_2 \leq r_f$ .

**Proof.** Consider the contrary, a fixation  $f$  has  $\ell_2 > r_f$  by Algorithm 4. A different  $\ell_2'$  can be calculated by randomly choosing exactly  $\mathcal{N}$  of the fixation points in  $f$ , as there

are at least  $\mathcal{N}$  fixation points in the box bounded by  $r_f$ . The enclosing square apothem can only decrease when reducing to  $\mathcal{N}$  of the enclosed points. Hence, we can conclude that  $\ell_2' \leq r_f$ . It suggests that these  $\mathcal{N}$  points have a smaller bounding box apothem  $\ell_2'$  than  $\ell_2$ , which contradicts the fact that  $\ell_2$  is the apothem of the minimum bounding box covering  $\mathcal{N}$  points in the given gaze data for fixation  $f$ . Hence, the original statement holds. ■

### Comparison of Two Lower Bounds

In this section, we discuss the relation between  $\ell_1$  and  $\ell_2$  and we find that neither bound dominates the other. Hence for  $r_f, f = 1, \dots, \mathcal{F}$ , we have  $\hat{\ell} \leq r_f$ , where  $\hat{\ell}$  is defined as one of  $\ell_1$  or  $\ell_2$ .

**Proposition 1** *Neither lower bound  $\ell_1$  or  $\ell_2$  dominates the other.*

**Example 1.** Consider the examples of identifying one fixation in a gaze sequence with seven points, as depicted in Figure 3.5. Supposing that  $\mathcal{N}$  is four and the outlier budget  $\mathcal{P}$  is one,  $\ell_1$  is determined by the  $x, y$  distances between  $\lfloor \frac{\mathcal{N}}{\mathcal{P}+1} \rfloor = \lfloor \frac{4}{2} \rfloor = 2$  consecutive points, while  $\ell_2$  is the apothem of the smallest square bounding box covering  $\mathcal{N} = 4$  points in the plane. The relationship of  $\ell_1$  and  $\ell_2$  varies based on the distribution of gaze points: (a) shows  $\ell_1 < \ell_2$ ; (b) shows  $\ell_1 = \ell_2$ ; and (c) shows  $\ell_1 > \ell_2$ .

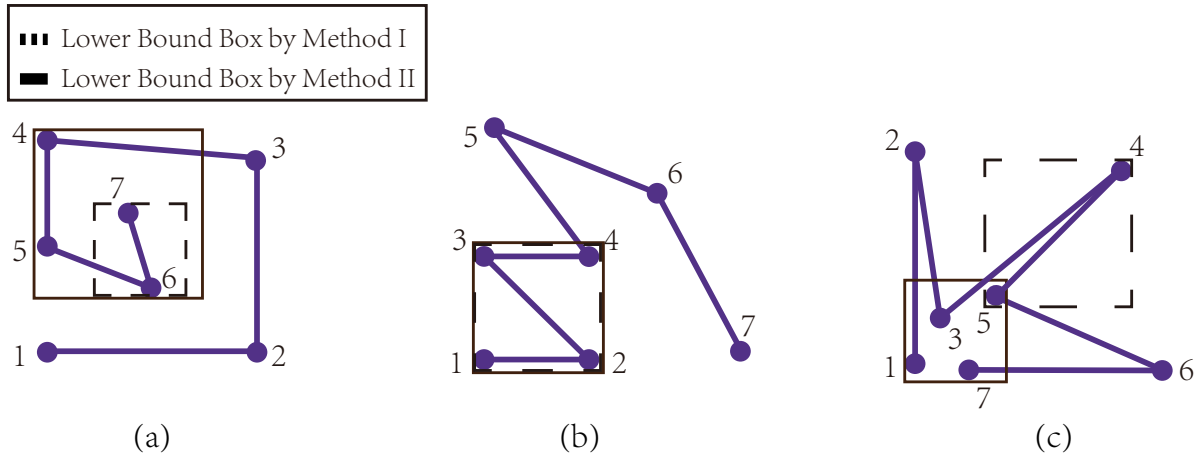


Figure 3.5: Comparison of lower bounding approaches. The gaze sequence length  $\mathcal{T} = 7$ , minimum number of covering points  $\mathcal{N} = 4$ , and outlier budget  $\mathcal{P} = 1$ . As shown in (a), (b) and (c), depending on how the points are distributed, the effectiveness of lower bounds  $\ell_1$  and  $\ell_2$  vary.

## 3.4 Computational Experiments

Formulation (3.4a)–(3.4j) with the decomposition principle described in Section 3.3.1 represents the FID<sup>+</sup> filter, which extends the earlier FID filter. We now discuss our computational experiments using real eye tracking data. We use a dataset obtained from the visual task of answering Graduate Record Examination (GRE) Math reading questions on a computer display in Chapter 1. Algorithms 3 and 4 are introduced to derive lower bounds on  $r_f$  to improve the computational performance for solving the new formulation.

### 3.4.1 Experimental Setup and Data Preprocessing

The GRE Math dataset contains ten recordings collected by a Tobii Pro TX300 eye-tracking device at 300 Hz. Each recording is approximately five minutes in duration. Table 3.1 summarizes this dataset. We used the same data preprocessing strategy as discussed in Chapter 1. For each recording, we separate the data sequence  $\mathcal{S}$  into chunks  $C^k$ ,  $k = 1, \dots, \mathcal{K}_\ell$  using the Tobii Studio I-VT filter [17] with the default velocity threshold of  $V = 30^\circ/s$ . The minimum number of gaze points is set to  $\mathcal{N} = 30$  (100ms), which is necessary for information processing to occur [13]. As shown in Table 3.1, this setting eliminates some data chunks and remain approximately 721 valid data chunks in each recording on average. We set  $\mathcal{F}_{min}^k = \mathcal{F}_{max}^k = 1$  for formulation (3.4a)–(3.4j). The fixation outlier budget  $\mathcal{P}$  is set as 1% of the total number of gaze points in each data chunk  $C^k$ , that is, outlier budget  $\mathcal{P}^k = \lceil 0.01 \cdot |\mathcal{C}^k| \rceil$ . This value of  $\mathcal{P}^k$  allows for at least one point per data chunk to be identified as a fixation outlier in formulation (3.4a)–(3.4j). As depicted in Figure 3.6(a), the distribution of data chunks is long-tailed. Of the total 7,208 data chunks with at least  $\mathcal{N}$  points, there are 1,860 data chunks having more than 100 points (25.8% of total), and 59 data chunks with length of greater than 500 points (0.8% of total). As the size of the data chunk increases, so does the expected computational effort in solving formulation (3.4a)–(3.4j). All computational experiments were conducted using an Intel core i7-6700MQ computer with 3.40 GHz and 16.0 GB RAM running 64-bit Windows 10. Gurobi Optimizer [30] with Python 2.7 was used for the optimization modeling, algorithm development and solution process. For each optimization problem, we use the default parameter setting of Gurobi MIPGap (1e-4) and MIPGapAbs (1e-10) for pursuing global optimality. We also set a time limit of one hour (wall-clock) for solving the optimization model for each data chunk. MATLAB 2016a [31] was used for additional data processing and analysis.

Stimuli	Avg # of All Points in Sequence	Avg # of Data Chunks	Avg # of Valid Data Chunks	Avg # of Points in All Data Chunks	Avg # of Points in Valid Data Chunks
GRE Math Reading Data	90,580	3,612	721	80,956	66,677

Table 3.1: Summary results on 300 Hz GRE Math Reading data with I-VT filter, averaged over ten recordings per dataset.

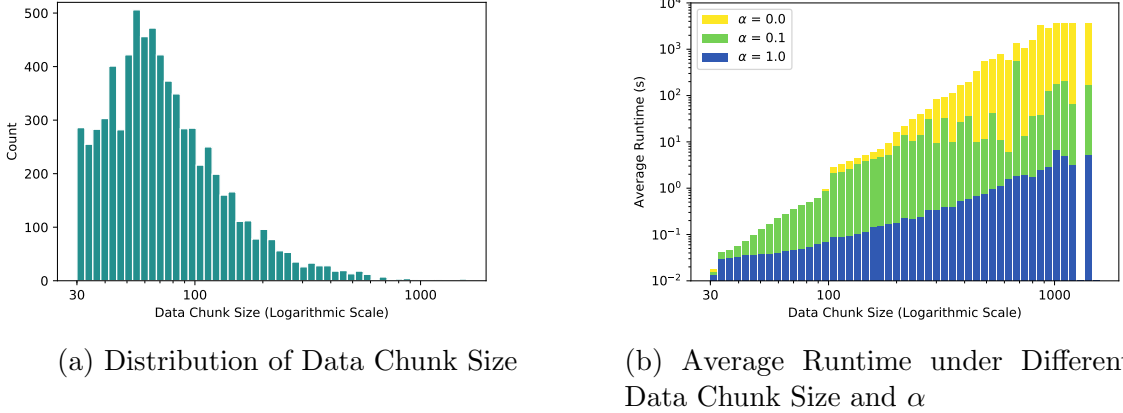


Figure 3.6: Depicting the distribution of data chunk size (left panel) and the average runtime using formulation (3.4a)–(3.4j) in each bin under  $\alpha = 0, 0.1, 1$  (right panel). The right panel also shows that with the increase of  $\alpha$ , the runtime decreases; with the increase of  $|\mathcal{C}^k|$ , the runtime increases substantially, and becomes especially apparent when  $|\mathcal{C}^k|$  exceeds 100.

### 3.4.2 Computational Results and Discussion

Table 3.2 highlights the computational results of running the FID<sup>+</sup> filter on the 300 Hz GRE Math reading dataset, as well as formulation (3.4a)–(3.4j) using lower bounds from Algorithms 3 and 4. The rows of Table 3.2 are indexed by parameter  $\alpha$ , and the columns display the evaluation metrics, budget usage and runtime, and are to be compared with those of Table 1.4 in Chapter 1, depicting similar results without outlier detection. As in Table 1.4, the evaluation metrics are averaged over all data chunks in each of the ten data recordings. The evaluation metrics we consider are: *fixation duration*  $\delta$ ; *cover rate*  $\gamma$ ; three fixation *inner-density* metrics:  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$ ; and *center shift*  $\lambda$ .

The average fixation duration  $\delta$  is the average number of fixation points in each fixation divided by the sampling frequency. The cover rate  $\gamma$  measures the ratio of points recognized as fixation points to the total number of points in a recording. We consider the three density metrics in Chapter 1. All of them are inversely proportional to density. That is, they represent greater density as the magnitudes get smaller. The first metric  $\rho_1$

is the average pairwise distances between fixation points within one fixation.

$$\rho_1 = \frac{\sum_{p=1}^{\mathcal{P}-1} \sum_{q=p+1}^{\mathcal{P}} d_{pq}}{\binom{\mathcal{P}}{2}}. \quad (\rho_1)$$

The second density metric  $\rho_2$  has the same numerator with  $\rho_1$ : the pairwise distances of all identified fixation points. The denominator is simply the number of fixation points. Hence, as the number of included points increases,  $\rho_2$  experiences greater amplification as compared to  $\rho_1$ . The reason that  $\rho_2$  is considered is due to the relationship with the objective function of the first formulation, *Minimize Average Intra-Fixation Sum of Distances* (formulation (1.10a)–(1.10f)). Though our demonstration for detecting fixation outliers focuses on the formulation (1.11a)–(1.11f), we retain  $\rho_2$  in our comparison for the sake of completeness.

$$\rho_2 = \frac{\sum_{p=1}^{\mathcal{P}-1} \sum_{q=p+1}^{\mathcal{P}} d_{pq}}{\mathcal{P}}. \quad (\rho_2)$$

The third density metric  $\rho_3$  is the minimal square area covering the fixation divided by the number of included fixation points.

$$\rho_3 = \frac{(2\hat{r})^2}{\mathcal{P}}. \quad (\rho_3)$$

The center shift  $\lambda$  measures the Euclidean distance between the FID fixation centroid to the I-VT filter centroid. Additionally, we report the fixation outlier budget usage  $\beta$ , which is the ratio of the total number of identified fixation outliers to the cumulative outlier budget over all data chunks in the ten data recordings. The reported runtime is the average of the cumulative runtime of all data chunks in each of the ten data recordings.

300 Hz GRE Math Reading Data													
$\alpha$	Duration	Density Measures			Cover Rate	Center Shift	Budget Usage	Avg Runtime (s)		Avg Runtime (s) w/ $\ell_1$		Avg Runtime (s) w/ $\ell_2$	
	$\delta^{avg}$ (s)	$\rho_1^{avg}$	$\rho_2^{avg}$	$\rho_3^{avg}$	$\gamma^{avg}$	$\lambda^{avg}$	$\beta$	Gurobi	Overall	Gurobi	Overall	Gurobi	Overall
0	0.1038	5.3981	81.2267	10.4529	0.2539	1.9494	0.91	12,548.8	12,647.9	10,624.0	10,724.8	10,729.2	10,969.9
0.1	0.2597	6.1667	231.2483	9.7190	0.6510	1.0814	0.87	1,898.2	2,006.3	1,637.5	1,742.5	1,560.6	1,802.7
0.2	0.2744	6.4515	259.4887	9.4059	0.6863	0.8331	0.82	315.5	430.6	242.3	345.5	249.3	493.9
0.3	0.2787	6.5700	268.5097	10.0599	0.6956	0.7397	0.75	186.3	305.6	145.9	253.6	150.9	398.1
0.4	0.2806	6.6383	273.4140	10.2574	0.6997	0.6916	0.74	147.1	267.0	116.3	226.0	115.6	363.2
0.5	0.2831	6.7417	279.7678	10.0763	0.7056	0.6213	0.45	128.8	247.6	96.3	205.4	99.7	347.4
0.6	0.2840	6.7941	282.5965	10.2516	0.7076	0.5861	0.41	108.9	225.9	82.3	191.0	83.9	331.7
0.7	0.2844	6.8136	283.7578	10.3622	0.7084	0.5750	0.41	97.7	214.7	71.8	181.6	73.0	320.3
0.8	0.2848	6.8364	284.9439	10.4752	0.7094	0.5603	0.39	88.4	205.7	63.1	172.5	64.5	311.6
0.9	0.2850	6.8465	285.3952	10.5318	0.7098	0.5541	0.38	80.3	197.3	56.9	164.8	58.4	306.7
1.0	0.2859	6.9006	288.2015	10.8704	0.7122	0.5151	0.24	73.4	190.7	51.2	158.4	52.9	303.1

Table 3.2: Results of the FID<sup>+</sup> filter, (3.4a)–(3.4j) with lower bound  $\ell_1$ , and (3.4a)–(3.4j) with lower bound  $\ell_2$  on 300 Hz GRE Math reading dataset. The entries in the evaluation metrics columns report the average metrics over all data chunks in each of the ten recordings; the entries in the runtime columns report the total runtime averaged over each each recording, containing approximately 721 data chunks.



Each entry in the evaluation metrics columns in Tables 3.2 and 1.4 is averaged over ten recordings and all data chunks per recording. Each entry in the runtime columns reports the averaged cumulative runtime for solving approximately 721 data chunks of the  $\alpha$ -densest fixations. For higher level of  $\alpha$ , e.g.  $\alpha = 0.8$ , the average runtime of each data chunk is less than 0.13 second. For the most time-consuming  $\alpha$  level,  $\alpha = 0$ , each chunk solved average of 17.8 seconds. In Table 3.2, all but twelve optimization models (eleven for  $\alpha = 0$ , and one for  $\alpha = 0.1$ ) solved to global optimality within the one-hour time limit for formulation (3.4a)–(3.4j). The addition of the lower bound  $\ell_1$  and  $\ell_2$  enabled two additional models at  $\alpha = 0$ , and the sole model with  $\alpha = 0.1$ , to be solved to global optimality.

The general trend of evaluation metrics and runtime from  $\alpha = 0$  to  $\alpha = 1$  are similar in Tables 3.2 and 1.4. It indicates that  $\alpha$  has a similar effect on fixation identification and fixation properties in both formulations.

When compared to Table 1.4, the entries in the initial columns of Table 3.2 demonstrate the effect of removing outliers. In particular, values of the average fixation duration  $\delta^{avg}$  rate are smaller in Table 3.2, indicating that less gaze points are identified as fixation points by the FID<sup>+</sup> filter. The difference of  $\delta^{avg}$  is actually rather small, roughly akin to a single gaze point, between Tables 3.2 and 1.4. Similar to  $\delta^{avg}$ , the average cover rate  $\gamma^{avg}$  value under every  $\alpha$  level is slightly smaller in Table 3.2. Both Tables 3.2 and 1.4 have the same increasing trends on  $\delta^{avg}$  and  $\gamma^{avg}$  when  $\alpha$  increases.

The three density metrics appear with smaller values in Table 3.2, as compared to Table 1.4. Recalling that density is larger for smaller values of  $\rho_1$ ,  $\rho_2$  and  $\rho_3$ , it demonstrates that when allowing outliers within fixations, the mathematical formulation can further refine gaze points to identify denser fixations. It is worth noting that  $\rho_3^{avg}$  is two to three times smaller in Table 3.2 than in Table 1.4.  $\rho_3^{avg}$  is the ratio of the minimal area bounding box of the identified fixation, to the number of points this fixation contains, is identical to the objective in formulation (3.4a)–(3.4j).  $\rho_3^{avg}$  becomes smaller either when the fixation bounding area is smaller, or when the fixation duration decreases.

This trend of  $\rho_3^{avg}$  is strong evidence for the impact of outlier points on fixation density. Using the outlier budget  $\mathcal{P}^k = \lceil 0.01 \cdot |\mathcal{C}^k| \rceil$  as specified in the experimental setup, 74.2% of the fixations by formulation (3.4a)–(3.4j) identify only a single outlier point per fixation (chunk size less than or equal to 100 points). This is further underscored in Table 3.2, as the change in fixation duration is relatively minimal. However,  $\rho_3$  reduced by nearly two thirds. This indicates that a small group of outlier points are substantially skewing the size of the minimum apothem  $r$  and so the minimum fixation bounding box, and should be eliminated in the fixation.

For all values of  $\alpha$ , the center shift  $\lambda^{avg}$  reported in Table 3.2 is larger than  $\lambda^{avg}$  in

Table 1.4;  $\lambda^{avg}$  measures the Euclidean distance (in pixels) between the FID fixation centroid (as specified by  $(x^f, y^f)$ ), and the I-VT filter centroid. This increase in  $\lambda^{avg}$  reflects stray data points being eliminated via the outlier budget in the FID<sup>+</sup> filter, so as to better concentrate around the actual fixation. The outlier budget ratio  $\beta$  in Table 3.2 decreases as  $\alpha$  increases, due to identified fixation outlier points being penalized in objective function (3.4a). Therefore, the penalty parameter  $\alpha$  not only serves for balancing the trade-off between density and number of fixation points in for formulation (3.4a)–(3.4j), it also has significant influence on the number of fixation outliers identified by the formulation. One notable finding is that the budget  $\mathcal{P}$  is not always used, even for small  $\alpha$  levels.

The improved fixation metrics come with the trade-off of increased computational run time. The Gurobi runtime in Table 3.2 increases substantially compared with Table 1.4. The increase appears between  $\alpha = 0$  and  $\alpha = 0.1$ , where much more effort is consumed in balancing the objective function trade-off of including a point, or incurring the penalty of  $\alpha$ . As shown in Figure 3.6(b), the average runtime at each level of data chunk size increases significantly at  $\alpha = 0$  and  $\alpha = 0.1$ .

The last four columns in Table 3.2 report the average Gurobi runtime and overall runtime when using lower bounds derived from Algorithms 3 and 4. Under all  $\alpha$  levels, the reported Gurobi runtime from formulation (3.4a)–(3.4j) with Algorithms 3 and 4 is less than the Gurobi time from solely solving the formulation (3.4a)–(3.4j), which demonstrates that the bounds produced by both of the algorithms are effective in reducing the computational difficulty to the solver. However, because Algorithm 4 requires additional computational cost for processing the dataset, the average overall runtime for formulation (3.4a)–(3.4j) with Algorithm 4 only outperforms the experiment using solely formulation (3.4a)–(3.4j) for the  $\alpha = 0$  and  $\alpha = 0.1$  levels. Moreover, the additional time cost for running Algorithm 4 averages around 246s. On the other hand, the time cost for running Algorithm 3 per chunk is negligible, and thus does not contribute to much additional time in Table 3.2. The average overall runtime of formulation (3.4a)–(3.4j) with Algorithm 3 is still smaller than the runtime for running the formulation (3.4a)–(3.4j) solely. The runtime comparison indicates that both of the algorithms contribute to reducing the runtime of solving optimization models. That said, because Algorithm 4 incurs additional computational cost for data processing, only formulation (3.4a)–(3.4j) with Algorithm 3 outperforms in both Gurobi optimization time and overall runtime at every  $\alpha$  level than only using formulation (3.4a)–(3.4j). Future work may focus on improving the computational efficiency of the implementation of Algorithm 4.

## 3.5 Conclusions

This chapter introduces outlier aware fixation identification for gaze data by extending the FID (fixation-inner-density) filter that identifies the densest fixations in gaze data. Our new FID<sup>+</sup> filter enables stray gaze points within fixations to be flagged and eliminated from fixation consideration, thereby increasing the accuracy and precision of key metrics relate to the actual fixation. Gaze data collected by eye-tracking devices is collected as sequence of points representing the locations where eyes focus. Spatially and temporally adjacent points are clustered as fixations. Fixation features – such as location, duration and inner-density – carry information about user attention and awareness in behavior research. Such features are inherently influenced by how fixations and saccades (gaze points between fixations) are labeled by the fixation identification algorithms. Downstream behavioral properties, such as dwell time, fixation heatmap and pupil dilation during fixations, are impacted by the accuracy and precision of the fixation identification approach that is used.

Two popular fixation identification methods in practice are the I-VT and I-DT filters. They use relatively simple properties of gaze data and can be implemented efficiently in commercial eye-tracking devices. However, they can lead to inaccurate fixation results, which will result in misrepresenting behavioral patterns. The FID filter in Chapter 1 overcomes the limitations of these baseline methods by integer optimization to optimize for fixation inner-density, with an iterative algorithm that exploits the ability to decompose an entire gaze stream into components, or chunks. In this chapter we augment the FID optimization formulation with a new set of variables that indicate whether gaze point  $t$  is an outlier for fixation  $f$ . Moreover, we carefully design enhanced constraints that enable the strict fixation time consistency condition to be relaxed, by allowing for a small budget of fixation outlier points to be admitted. The enhanced integer optimization formulation (3.4a)–(3.4j) can recognize stray gaze points as fixation outliers, a concept that is underexplored in fixation identification algorithms. Raw gaze data contains inevitable noise (as depicted in Figure 3.3(b)), and we demonstrate that the FID<sup>+</sup> filter outlined in this chapter can robustly identify within-fixation outlier points, which is a significant enhancement to the existing FID filter in Chapter 1.

We conduct computational experiments to compare the new FID<sup>+</sup> filter with the FID filter with formulation (1.11a)–(1.11f) on the 300 Hz GRE Math reading dataset used in Chapter 1. The result shows that the FID<sup>+</sup> filter can identify fixations with substantially greater density. In particular, when comparing the density metric  $\rho_3$ , the ratio of minimal area bounding box and fixation point number, the FID<sup>+</sup> filter featured a 2-3 times reduction in  $\rho_3^{avg}$  while considering a small number of points as outliers within each fixation. Thus, these developments hold much promise for outlier-aware fixation

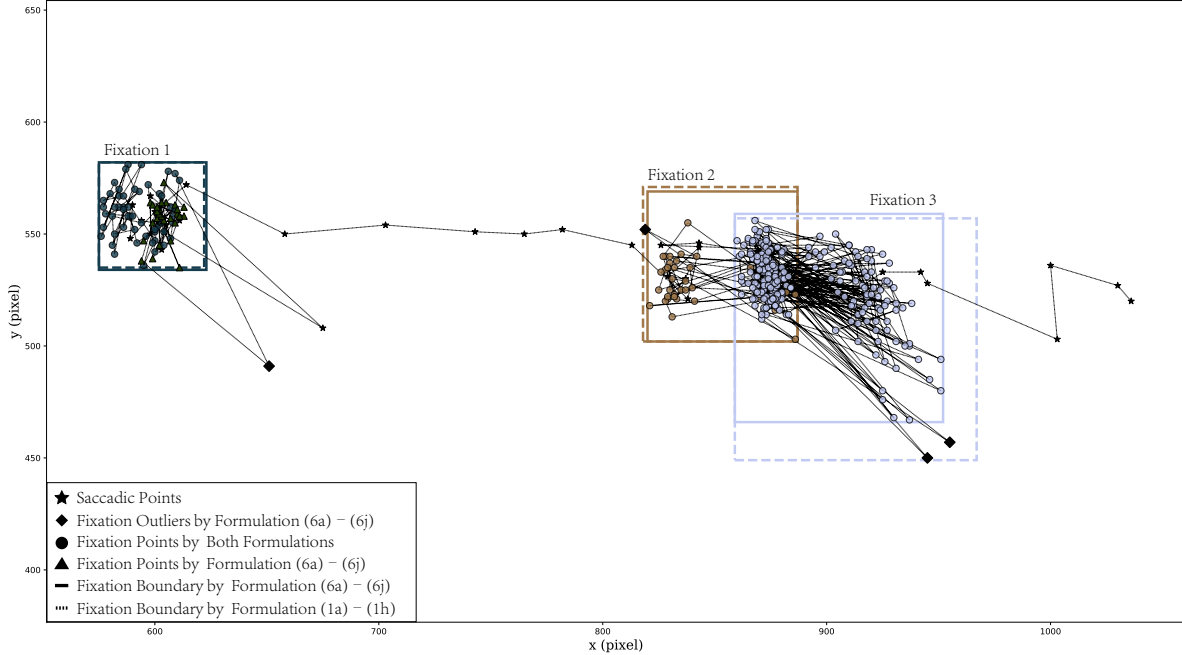


Figure 3.7: Fixation identification result with the  $FID^+$  filter versus the  $FID$  filter,  $\alpha = 0.5$ , on the gaze sequence in Figure 3.3(b).

identification.

Figure 3.7 highlights the comparison of fixation identification results from the  $FID^+$  filter and the  $FID$  filter. The illustrated gaze stream segment contains three fixations. For Fixation 1, while the identified fixation boundary looks identical for both methods, it turns out that, due to the ability to eliminate outlier points, the enhanced formulation contains 50% more points than the original formulation. This has the unexpected effect that formulation (3.4a)–(3.4j) has a slightly larger area, because such increased area greatly increases the number of included fixation points after outlier removal. Formulation (1.11a)–(1.11f) identifies all gaze points appearing before the outlier point flagged by formulation (3.4a)–(3.4j) as non-fixation points, while balancing the inherent trade-off present in objective function (3.4a). The gaze points at Fixation 2 are well clustered, so the two formulations have fairly similar results. For Fixation 3, formulation (3.4a)–(3.4j) identifies two fixation outliers and the fixation area decreases significantly as compared with the area identified by formulation (1.11a)–(1.11f). The outlier-aware identification results of formulation (3.4a)–(3.4j) likely have substantial impacts on the number of identified fixation points, as well as fixation bounding regions. This behavior is similar across chunks in the gaze data stream.

The approach outlined in this chapter does have some limitations. Due to the additional variables and constraints, the runtime for solving formulation (3.4a)–(3.4j) is slower than formulation (1.11a)–(1.11f) at each level of  $\alpha$ , and substantially so for  $\alpha = 0$  and

$\alpha = 0.1$ . We introduce two geometric arguments, and algorithms, for deriving lower bounds on  $r_f$  to accelerate the speed of reaching global optimality. Both algorithms find stronger lower bounds ( $\ell_1$  and  $\ell_2$ ) that are able to reduce Gurobi runtime, although more work is needed to improve the competitiveness for a small number of instances at  $\alpha = 0$  and  $\alpha = 0.1$ . Moreover, more work remains for refining Algorithm 4 to reduce its overall run time for computing lower bound  $\ell_2$ . Another possible direction of future work is to more carefully investigate suitable budget values for each data chunk. While we set the outlier budget value to approximately 1% of the length of the data chunk, other features such as data chunk dispersion, and the average velocity of points, could suggest improved estimates for the number of fixation outliers. Each data chunk could thereby have a data-driven budget value based on its features.

# Chapter 4

## Exploratory Data Analysis for Recommending $\alpha$ to the FID Filter users

The optimization-based formulations in Chapters 1 through 3 are all parametrized by the density parameter  $\alpha$  that enables decision-makers to have fine-tuned control over the density. There is significant opportunity for recommending suitable levels for  $\alpha$  to users. In this chapter, we first describe a manual method for creating two labeled datasets that leverages an interactive tool developed specifically for this purpose. We discuss our exploratory data analysis of the suitable  $\alpha$  levels for the data chunks in these datasets based on statistical measurements. We then explore the development of a machine learning model to automate the assignment of  $\alpha$  based on features of gaze data, which will subsequently be fed into the optimization formulations. After, we run the FID filter with the *Minimize Square Area of Fixations* formulation from Chapter 1 for various levels of  $\alpha$  on the training dataset, recording the  $\alpha$  level(s) providing an outcome most closely resembling the labels. Next, we extract various features of the data. These features are used to predict a suitable  $\alpha$  level for the optimization models, thereby eliminating the effort required to manually adjust  $\alpha$ . The final validation step demonstrates the model performance on the test dataset. We conclude our current findings and discuss potential directions for future work and improvements.

### 4.1 Introduction

Identifying fixations in gaze data is similar to the problem of determining sensible clusters, which is known to be subjective in nature and difficult to evaluate. One common approach to evaluate fixation identification algorithms is by comparing the algorithmic results with


the event detection results obtained from eye-tracking experts. Comparing the algorithmic performance with that of a human, is a check on its effectiveness. Moreover, fixation identification algorithms typically incorporate parameters or thresholds determined by users during the process. Tuning parameters requires some domain knowledge about fixation identification, such as the angular velocity of eye movement, that is, how the eye tracker projects gaze onto the 2D plane. A small subset of those parameters might have substantial influence on identification result, which makes the choice of such key parameters both critical for algorithmic performance, and challenging to estimate by hand. Deciding what parameters should be used under different eye-tracking experimental environment is cumbersome to users, especially to novice users, who may not know what values to use by default. The ability to suggest reasonable parameter settings is thus attractive to most users. Measuring the performance agreement with human experts while tuning parameters can provide useful information about suitable parameters. Inspired by [34, 41], we label a portion of gaze data by human experts in eye tracking research. The computational performance of the FID filter under various  $\alpha$  levels can thereby be evaluated by comparing with the labels marked by human experts. In addition to fixation and density metrics, we explore the implementation of supervised learning methodologies to recommend suitable  $\alpha$  levels for users of the FID filter.

## 4.2 Dataset and Equipment

In Section 1.4.2, we introduced the 300 Hz GRE Math dataset that contains the eye tracking recordings of ten participants. For each recording, we sample ten subsequences, each of which contains 1,000 points. We also study a second dataset containing 47 eye-tracking recordings at 300 Hz from different participants under an Online Shopping task. The visual stimulus is a static webpage (Figure 4.1) showing the items and their attributes in a tabular arrangement. The participants are asked to choose the best item they would like to purchase after comparing the item information. We sample one hundred subsequences from the online shopping recordings, with each subsequence containing 1,000 points. Each recording is at least sampled once.

Both of the datasets were collected from the User Experience and Decision Making (UXDM) lab at WPI. We employed three expert eye tracking researchers to manually identify whether gaze points in the randomly sampled sequences are categorized as fixations (labeled as 1) or saccade points (labeled as 0).

### Laptops



Attribute	Laptop A	Laptop B	Laptop C	Laptop D	Laptop E
<b>Black Friday Price</b>	<del>\$999</del> \$849	<del>\$1499</del> \$1299	<del>\$1099</del> \$999	<del>\$749</del> \$699	<del>\$1099</del> \$899
<b>Processor</b>	Core™ i5 Processor 2.2GHz ★★★★ 4	Core™ i7 Processor 3.2GHz ★★★★★ 5	Core™ i5 Processor 2.6GHz ★★★★ 4	Core™ i3 Processor 1.2GHz ★★★ 3	Core™ i5 Processor 2.6GHz ★★★★ 4
<b>RAM</b>	8GB ★★★★ 4	16GB ★★★★★ 5	16GB ★★★★★ 5	8GB ★★★★ 4	8GB ★★★★ 4
<b>Drive</b>	128GB SSD ★★★ 3	128GB SSD + 1T HDD ★★★★★ 5	500GB HDD ★★★ 3	128GB SSD ★★★ 3	256GB SSD ★★★★★ 5
<b>Graphic</b>	MX150 with 2G ★★ 2	GTX 1060 with 6G ★★★★★ 5	GTX 1050 Ti with 4G ★★★★ 4	Intel HD 630 ★ 1	GTX 1050 with 4G ★★★ 3
<b>Screen</b>	13.3" (1920 x 1080)	15.6" (1920 x 1080)	13.3" (1920 x 1080)	11" (1920 x 1080)	13.3" (1920 x 1080)
<b>Touch Screen</b>	Yes	No	No	Yes	No
<b>Weight</b>	4.1lb	6.9lb	5.1lb	2.3lb	4.9lb
<b>Battery Life</b>	7.6hrs	4.4hrs	4.1hrs	9.6hrs	5.8hrs
<b>Customer Rating</b>	★★★ 3	★★★★★ 4	★★★☆☆ 3.5	★★★☆☆ 3.5	★★★★☆ 4.5
	<input type="button" value="Add to Cart"/>	<input type="button" value="Add to Cart"/>	<input type="button" value="Add to Cart"/>	<input type="button" value="Add to Cart"/>	<input type="button" value="Add to Cart"/>

Figure 4.1: The webpage for the Online Shopping task in the eye-tracking experiment.

## 4.3 Data Labeling Toolbox

We develop a toolbox using Python 3 and Tkinter, a standard GUI (Graphical User Interface) package. Figure 4.2 shows our data labeling toolbox panel. A user first chooses a .csv file with a  $(x, y)$  pair at each line. The labeling program goes through the gaze sequence point by point. After clicking *Generate Graphs*, the toolbox displays four panels for user to review the gaze point category. These panels provide different information for users to inspect each gaze point. The upper left panel shows the current gaze point by highlighting it as red, with the trajectory of the next 50 points. The upper right panel shows previously labeled fixation and saccade points. The lower left panel shows the gaze point velocity and the lower right panel presents the whole path of the gaze sequence. Users can mark the current gaze point as a fixation point or a saccadic point and then click the *confirm* button and continue to label the next gaze point. The user also can go back to previous labeled gaze point by point index navigation, if desired, to make some changes.



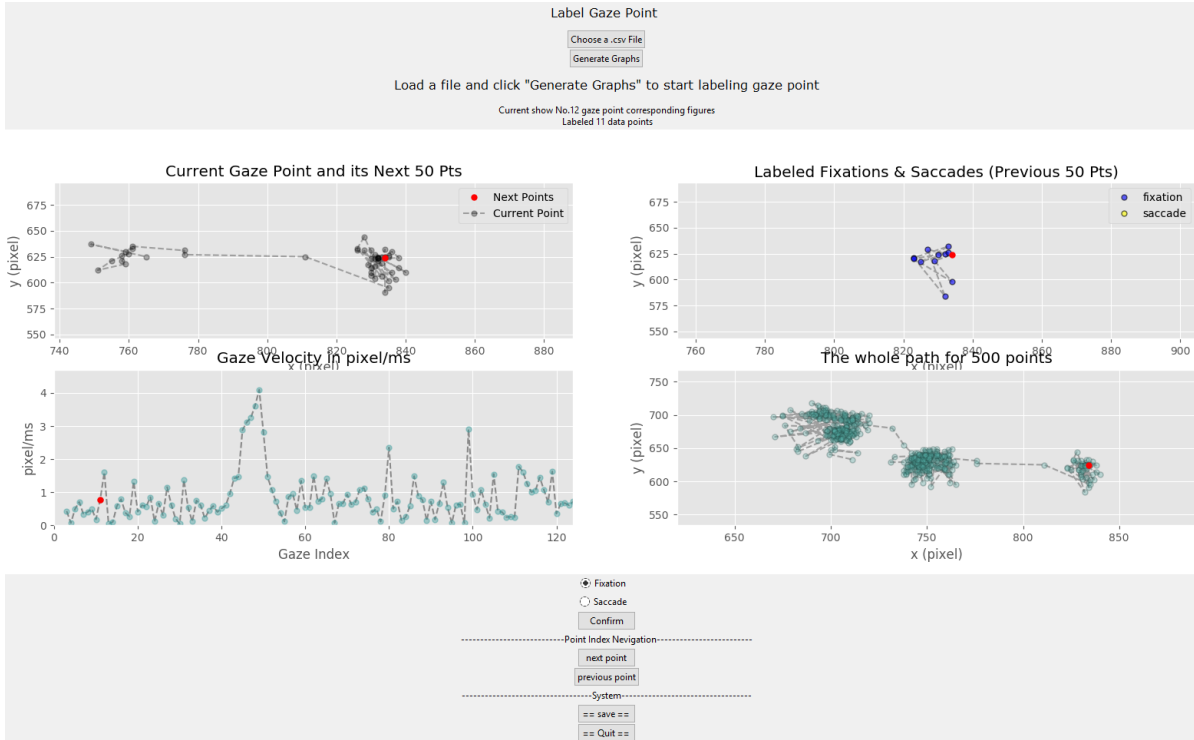


Figure 4.2: Data Labeling Toolbox Panel.

## 4.4 Exploratory Data Analysis

In Section 1.4.2, we described the data preprocessing method to separate data sequence into data chunks. Each data chunk is the input data for our optimization approach and may have properties that can suggest an appropriate alpha setting. Because we randomly select the beginning of the gaze point sequence from each data recording, some labeled gaze points belong to data chunks that are not fully enclosed in the labeled data. Within the 200 thousand labeled gaze points, we distilled 865 data chunks from the GRE Math dataset and 1,206 data chunks from the Online Shopping dataset. Each gaze point from each of these chunks is manually labeled according to expert opinion. We find that the classes of fixation points and saccade points are highly imbalanced in both of the GRE Math and Online Shopping recordings. Of all the labeled gaze points, approximately 90% points are fixation points, which might be because the visual stimulus are static in both eye-tracking tasks. Our finding is consistent with the description in [41].

For each data chunk, we run formulation (1.11a)–(1.11f) for  $\alpha$  with the range from zero to a number large enough so that the identified fixation includes all the data chunk points. That is, when  $\alpha$  is large enough, formulation (1.11a)–(1.11f) has the same identification result with the I-VT filter used for data preprocessing. For the sampled data chunks from the GRE Math dataset, we find  $\alpha = 50$  is such a number for the formulation that all the identified fixations are identical with the data chunks, whereas  $\alpha = 22$  is large enough for

all the data chunks from the Online Shopping data. Thus, we run  $\alpha$  from 0 to 50 and 0 to 22, respectively, on the two types of data chunks. The step size is set as 0.1. We consider the  $F_1$  score as the measure for fixation identification performance because it accounts for both the precision and the recall in classification result.

For each data chunk, we record all of the  $\alpha$  levels that can reach the maximum  $F_1$  score by comparing with the fixation point labels in this data chunk. The preprocessing procedure already filtered most of the saccade points, so the number of points labeled as saccade points in the data chunks is quite low. The  $F_1$  score at appropriate  $\alpha$  levels is particularly high for measuring fixation point labels; many of the largest  $F_1$  scores are greater than 0.98. One interesting finding is that the  $\alpha$  levels with the highest  $F_1$  scores are consecutive, e.g.,  $\alpha = 0.1, 0.2, \dots, 3.0$ . We name this range of  $\alpha$  levels as the *optimal range*. We also find that 82.3% of the data chunks have the highest  $\alpha$  level (respectively,  $\alpha = 50$ , or  $\alpha = 22$ ) in their optimal  $\alpha$  range. It indicates that categorizing all points as fixation points in those data chunks best matches with the labels of fixation points measured by the  $F_1$  score. Thus, implementing the FID filter on those data chunks could not further enhance the fixation identification performance because when  $\alpha$  is large enough, the fixation identified by Formulation (1.11a)–(1.11f) is precisely identical to the data chunk. Hence, the optimal  $\alpha$  range for those data chunks is actually left-bounded and right-unbounded. That said, for the data chunks having a right-bounded optimal  $\alpha$  range, Formulation (1.11a)–(1.11f) is effective to refine the data chunks. Figure 4.3 shows the histogram of the mean value of their optimal  $\alpha$  ranges. The histogram is right-skewed, which illustrates a few values are much larger than the rest.

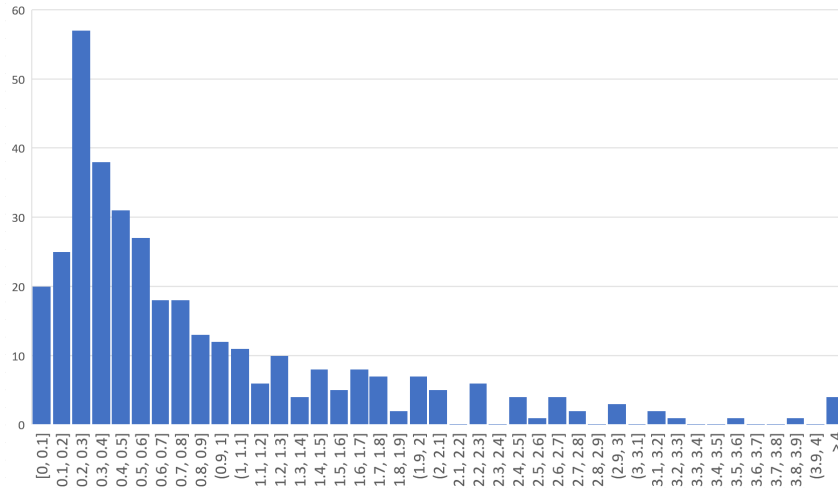


Figure 4.3: Histogram of the mean value of the optimal  $\alpha$  range for the data chunks needing further refinement by Formulation (1.11a)–(1.11f).

## 4.5 Predictive Modeling

Based on these observations, we could conclude that recommending a suitable  $\alpha$  value for processing the remaining 17.7% of the data chunks carries actual meaning for increasing fixation identification agreement with human experts. Therefore, we explore a two-step machine learning model for automatically tuning  $\alpha$  for Formulation (1.11a)–(1.11f) based on the labeled data. The first step is to build a binary classifier using the data chunk features to recognize whether an input data chunk needs to be refined by Formulation (1.11a)–(1.11f). The second step is to build a regression model to predict the mean  $\alpha$  of their optimal range for those data chunk needing for further refinement.

### 4.5.1 Training and Testing Datasets

We randomly select 30% of participants and their eye-tracking recording in each of the datasets to formulate a testing set for final performance evaluation of the machine learning model. Three participants with 30 subsequences in the GRE Math dataset and 14 participants with 17 subsequences from the Online Shopping dataset are held for testing. The testing set contained 655 data chunks. The rest of recordings will be used as training data for selecting the learning models and parameter tuning with cross-validation. We use Scikit-learn [42], a Python module that provides state-of-the-art machine learning algorithms for the predictive analysis.

### 4.5.2 Feature Extraction

We generate 28 representative features for each data chunk. Based on the feature extraction described in [43], the statistical features of gaze location, gaze velocity, and fixation duration are extracted from the data chunks. The MIP formulation (1.11a)–(1.11f) we are implementing in the FID filter optimizes for the apothem of fixation bounding square, which is also monotonically with the density metric  $\rho_3$ : the minimal area square bounding box divided by the number of points within a data chunk. We calculate the apothem of the bounding box and  $\rho_3$  of the data chunk as the features, as it is believed that they may be correlated to the fixation apothem and  $\alpha$ . As formulation (1.11a)–(1.11f) always classify points at the beginning and ending location in a data chunk as saccade points when using the experiment setting in Section 1.4.2, the velocity of the beginning and ending points in a data chunk can be indicative to suitable  $\alpha$  levels. We thus extract an additional six features to describe their velocity. All of the generated features are listed in Table 4.1.

No.	Features	Description
1-4	Statistical features of gaze point velocity in X direction	Mean, standard deviation, skewness, kurtosis of velocity
5-8	Statistical features of gaze point velocity in Y direction	Mean, standard deviation, skewness, kurtosis of velocity
9	Data chunk duration	Number of points divided by the sampling frequency
10	Standard deviation ( $X$ )	Standard deviation of $X$ s
11	Standard deviation ( $Y$ )	Standard deviation of $Y$ s
12	Skewness ( $X$ )	Statistical skewness of $X$ s
13	Kurtosis ( $X$ )	Statistical kurtosis of $X$ s
14	Skewness ( $Y$ )	Statistical skewness of $Y$ s
15	Kurtosis ( $Y$ )	Statistical kurtosis of $Y$ s
16	Path length	Total length of scanpath for a data chunk
17	Dispersion	Spatial spread for a data chunk: $D = (\max(X) - \min(X)) + (\max(Y) - \min(Y))$
18	Dispersion of X direction	Spatial spread for a data chunk: $D_x = (\max(X) - \min(X))$
19	Dispersion of Y direction	Spatial spread for a data chunk: $D_y = (\max(Y) - \min(Y))$
20	Average velocity	Average amplitude of gaze movement velocity
21	Radius of square bound box	Minimum radius for a square bound box to cover all points in a data chunk
22	Density metric	The minimal area square bounding box divided by the number of points within a data chunk
23	Velocity of the first gaze point	The amplitude of the first gaze point movement in a data chunk
24	Velocity of the second gaze point	The amplitude of the second gaze point movement in a data chunk
25	Velocity of the last gaze point	The amplitude of the last gaze point movement in a data chunk
26	Velocity of the second-to-last gaze point	The amplitude of the second-to-last gaze point movement in a data chunk
27	Average velocity of the first five gaze points	The amplitude of the first five gaze point movement in a data chunk
28	Average velocity of the last five gaze points	The amplitude of the last five gaze point movement in a data chunk

Table 4.1: List of the 28 features generated for each labeled data chunk (note: top two lines contain 8 features).

### 4.5.3 Step One: Classification Model

The first step of our predictive analysis is to build a binary classifier for recognizing the data chunks needed for further refinement by Formulation (1.11a)–(1.11f). We label these data chunks as positive class and the others as negative class. As described in the beginning of Section 4.5, the sample size in the positive class is somewhat smaller than the negative class. The scale of the two classes is about 1:4.6. Thus, balancing the two classes is necessary before training the machine learning models. We perform Synthetic Minority Over-sampling Technique (SMOTE) [44], a popular method to over-sample the minority class. The sample size ratio is balanced to 1:1 for training. We then evaluate the performance by nested cross-validation (CV) on a group of commonly used machine learning models: Support Vector Machine, Logistic Regression, Random Forest, Gradient Boosted Trees, and XGBoost. Nested CV performs a series of CV. The inner CV is used for training the models and optimizing the hyperparameters. The outer CV is used for final model selection. By this procedure, nested CV effectively avoids overfitting and data snooping. Our performance metric is the area under the Receiver Operating Characteristic curve (AUC) score from prediction scores for tuning hyperparameters and model selection. The performance evaluation result shows that the Random Forest classifier reaches the highest AUC score at 0.962 on the balanced training set with the parameter setting as 170 trees and maximal depth as 10.

In the testing dataset, 95 data chunks are in the positive class and 560 data chunks are in the negative class. Figure 4.4 shows the confusion matrix of predicting the labels of data chunks in the testing set. For the positive class, the precision is 0.45 and the recall

is 0.54. Figure 4.5 shows the feature importances from the Random Forest classifier. One interesting finding is that the most important feature is the statistical kurtosis of gaze location in  $X$ -axis. The second and third important features are the average velocity of the first five gaze points and the velocity of the first gaze.

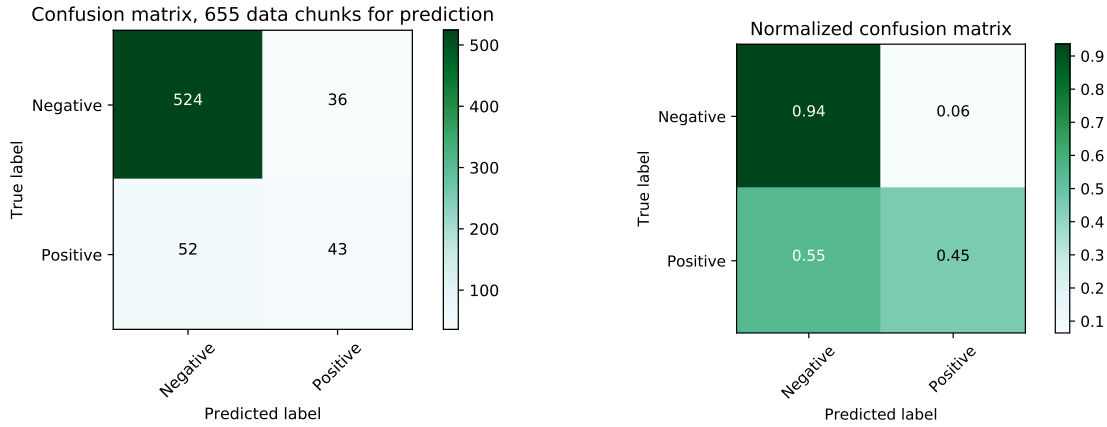


Figure 4.4: The confusion matrix of predicting data chunk categories in the test dataset.

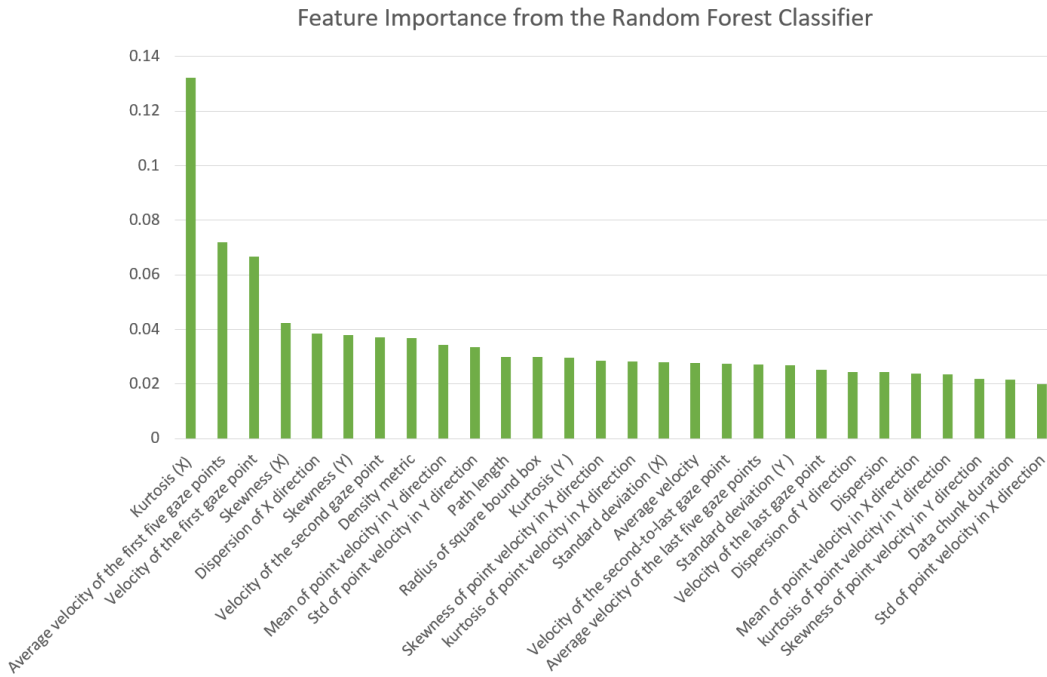


Figure 4.5: Feature importances from the Random Forest classifier. The top three most important features are: kurtosis ( $X$ ), average velocity of the first five gaze points, velocity of the first gaze point.

#### 4.5.4 Step Two: Regression Model

For fitting a regression model of those data chunks that need to be further refined by formulation (1.11a)–(1.11f), we implemented the Huber Regressor [45], a linear regression

model that is robust to outliers. The Huber Regressor minimizes a hybrid of linear loss for the samples that are classified as outliers, and squared loss for the other samples. We use CV to optimize the regularization parameter while fitting the regression model. The model reaches a minimum mean squared error when the regularization parameter is 1.0. The coefficient of determination ( $R^2$ ) is 0.568 when predicting mean of optimal  $\alpha$  levels on testing data chunks. There are 59 predictive values of  $\alpha$  that lie in the observed optimal  $\alpha$  range.

## 4.6 Findings and Discussions

We cascade the trained Random Forest Classifier and Huber Regressor as a two-step predictive model: the first step is using the Random Forest Classifier to predict the class of an input data chunk; if this data chunk is classified as needing refinement, the regression model in the second step predicts an  $\alpha$  level for implementing formulation (1.11a)–(1.11f). On the testing dataset, the classifier filters out 79 data chunks (43 true positive chunks) for further processing. In the second step, 45 data chunks get a predicted  $\alpha$  from the regression model lying in their actual optimal  $\alpha$  range. The correct predictions for the negative class in the first step can be viewed as obtaining a significantly large  $\alpha$  value in the optimal  $\alpha$  range that leads the FID identification result to be the same as the input data chunk. The predictive model correctly recommends an  $\alpha$  value to 569 data chunks.

Adjusting parameters for fixation identification algorithms is typically required for each individual eye-tracking experiment. When considering the general accuracy for identifying fixation points within data chunks via the predicted  $\alpha$  levels with formulation (1.11a)–(1.11f), we calculate an overall  $F_1$  score for all the subsequences from each participant in the test dataset. The result shows that for the GRE Math dataset, the average  $F_1$  score over participants obtained by the formulation (1.11a)–(1.11f) with predicted  $\alpha$  values ranks at the first place with the value of 93.614%. The second highest average  $F_1$  score, 93.607% is from running formulation (1.11a)–(1.11f) with a fixed  $\alpha$  in the range of [1.6, 1.9]. The third highest  $F_1$  score is obtained by  $\alpha$  from 4.8 to 5.9. The lowest  $F_1$  scores are at  $\alpha = 0$  and  $\alpha = 0.1$ , with the value of 56.050% and 88.860%. When examining the Online Shopping dataset, the highest average  $F_1$  score is 98.176%, which is reached by  $\alpha$  from 8.5 to 21. The formulation (1.11a)–(1.11f) with predicted  $\alpha$  values has the  $F_1$  score as 98.172%.

When comparing the  $\alpha$  levels with the highest  $F_1$  score on these two types of eye-tracking data, we find that the optimal  $\alpha$  range is actually different. The GRE Math dataset prefers  $\alpha$  at smaller levels whereas the Online Shopping data favors higher  $\alpha$  levels. The identification result by the formulation (1.11a)–(1.11f) with the different

predicted  $\alpha$  values outperforms running the formulation with any fixed  $\alpha$  level. However, due to the limited GRE Math participants in the test data, we could not draw the conclusion that this result is statistically significant. For the performance comparison on the Online Shopping dataset, the average  $F_1$  score gained from experiment with different predicted  $\alpha$  values is slightly lower than running with fixed  $\alpha$  values. To evaluate if the  $F_1$  score with the predicted  $\alpha$  values is significantly smaller than the highest  $F_1$  score obtained by the fixed  $\alpha = 8.5$  for all the participants, we ran a paired  $t$ -test on the 14 participants for the Online Shopping task in this testing data set. The  $p$ -value is 0.92, indicating that the  $F_1$  score by the predicted  $\alpha$  values is not significantly smaller.

## 4.7 Conclusions

This chapter introduces our findings for comparing the fixation identification results by the FID filter with the *Minimize Square Area of Fixations* formulation on the eye-tracking datasets with fixation and saccade points labeled by human experts. We demonstrate the process for creating labeled datasets: the GRE Math dataset and the Online Shopping dataset. We implement the FID filter that incorporates the data preprocessing using the I-VT method to separate original data records into data chunks. Then, the optimization formulation (1.11a)–(1.11f) parametrized over a range of density modulation parameter values  $\alpha$  is run to refine the I-VT identification result in each data chunk.

We first discuss our exploratory data analysis for recommending  $\alpha$  at a micro level – for each data chunk, we evaluate the identified fixation points with the human expert labels. In the two labeled datasets, we find that 82.3% of the data chunks with their gaze points are labeled as fixation points by human experts. When running the FID formulations to refine the data chunk points, it might be more efficient to filter out those data chunks and directly output them as fixations. We built a Random Forest classifier to recognize those data chunks by the generated features and demonstrate the classification performance. For the remainder of the data chunks, we find that the optimal  $\alpha$  levels as determined by the optimization performance lie in a range of values. That is, each data chunk has an optimal  $\alpha$  range for the FID filter that leads to the identified fixation points most agreeing with human expert labels. We explore a Huber Regressor model, a linear regression model that is robust to outliers, to predict the mean of their optimal  $\alpha$  range from the dataset.

We then discuss our exploratory findings at a more general level – for each eye tracking recording in the dataset, we compare the identification results by the formulation (1.11a)–(1.11f) with different  $\alpha$  levels, with the human labeled results. We find that the GRE Math dataset prefers  $\alpha$  at smaller levels than the Online Shopping data, by observing random

sampled participant eye tracking recordings. We may recommend  $\alpha$  levels around 1.75 for the GRE Math eye-tracking recordings and for similar scenarios. For the Online Shopping dataset, we would recommend  $\alpha$  levels over 8.5. When comparing the overall performance between running formulation (1.11a)–(1.11f) with the predicted  $\alpha$  levels and with each single  $\alpha$  level, we find that the result from the predicted  $\alpha$  levels ranks at the highest level as measured by the average  $F_1$  score. However, perhaps due to the limited number of testing participants, we are unable to draw the conclusion with statistical significance. For the Online Shopping dataset, the average  $F_1$  score by the predicted  $\alpha$  levels ranks lower than the results from several fixed  $\alpha$  levels. We perform a paired  $t$ -test to see if the difference is significant; the result shows that the average  $F_1$  score is not significantly smaller than the scores by fixed  $\alpha$  levels.

The discussions for recommending  $\alpha$  levels outlined in this chapter have some limitations. As mentioned in Section 4.4, we find that about 90% of the gaze points are as fixation points. In both of our labeled datasets, the preprocessing procedure by the I-VT filter eliminates most of the saccade points, so the class of gaze points in the data chunk is highly imbalanced – nearly entirely points are fixation points. It leads to the  $F_1$  score is exceptionally high when measuring fixation points. Therefore, it may be necessary to find other proper performance metrics for evaluation purposes. Thus far, we could not draw a definitive conclusion that our two-step predictive model performs better than when using a fixed  $\alpha$  while running formulation (1.11a)–(1.11f). Another idea is to label more gaze points in the currently used dataset.

Our discussions are all based on the sample-by-sample comparison with the human expert labels. An additional research direction is the event-by-event comparison of fixations, that is, to compare the duration and location of the identified fixations with the labeled fixations. Moreover, we only used formulation (1.11a)–(1.11f) for discussing optimal  $\alpha$  levels. We may further adapt the formulation (3.4a)–(3.4j), *Minimizing Square Area of Fixations with Outlier Sensitivity* in our experiment and evaluate the performance on labeled datasets.



# Chapter 5

## Conclusions and Future Work

The major theme of this dissertation research is the application of mathematical optimization techniques and associated algorithms to identify fixations in eye gaze data. The accurate classification of eye gaze data into its constituent components is a key factor in *behavioral studies*, as eye gaze data forms the foundation for ensuing information processing. Gaze data is commonly categorized into two main events: *fixations* are clusters of points that are near in proximity and time, whereas *saccades* are higher velocity gaze points that occur between fixations. The majority of behavioral analyses focus on studying *global* fixation patterns. The distribution of gaze points within an individual fixation, which we call *micro-patterns*, has thus far been largely ignored. Prior work in [10] shows that these patterns can reveal significant information about focused attention and effort, which our findings further support. This forms a key milestone of this research: optimizing the classification of gaze data points using the notion of *inner-density*.

Fixation inner-density combines both the temporal and the spatial aspects of a fixation. These aspects are combined to form a measure of compactness of a fixation, which reveals significant and previously undiscovered information about attention. Inner-density can also overcome several limitations of existing methods, such as lack of sensitivity to peripheral points of a fixation, as well as the misrepresentation of fixation properties. We use integer optimization techniques to identify fixations in a sequence of gaze points by optimizing for inner-density, which we call the *FID filter*. A key novelty is the guarantee that there is no better gaze point identification according to the objective function of maximizing inner-density. While optimizing the entire data stream is computationally prohibitive, by exploiting the fact that saccades are natural separators of fixations, the entire gaze stream can be decomposed into a series of chunks, which enables efficient processing. Computationally speaking, extensive testing on real datasets demonstrates that our optimization-based approach is efficient and effective, identifying densest fixations in chunks in less than one second, on average. The identified fixations exhibit greater

density than existing methods, reflecting the ability of our approach to refine fixations, as well as more accurately represent gaze metrics such as fixation duration and center. The refined, denser fixations better represent attention and awareness for further analysis in eye tracking studies.

Building upon this initial milestone, in Chapter 2 we conduct extensive statistical testing of the FID filter with a widely used method of fixation identification, namely the *I-VT filter*, on a Text Reading eye tracking dataset. The results show that in general, fixations identified by the FID filter are significantly denser and more compact around their fixation center. They are also more likely to have randomly distributed gaze points within the square box that spatially bounds a fixation. The results of this study suggest that the FID filter increases the sensitivity of grouping gaze points into dense fixations, which are better representations of user focused attention in eye tracking investigations.

In Chapter 3, we extend the mathematical optimization models in the FID filter to account for fixation *outlier sensitivity*. The enhanced mathematical optimization models for the FID filter, which we call the *FID<sup>+</sup>* filter, improve the fixation identification by eliminating a small portion of fixation outliers. These conditions are captured by extending the current mathematical model to incorporate additional variables and constraints. While the addition of this extra flexibility introduces some additional complexity, we propose and implement approaches to improve the computational performance, by developing two arguments for tightening the lower bound of minimizing the square area of fixations.

The mathematical optimization formulations in our FID filter are all parametrized by a unique, manually assigned parameter  $\alpha$  that enables decision makers to have fine-tuned control over the density. The  $\alpha$  levels may have significant influence on the fixation identification results by the FID filter. In Chapter 4, we conduct exploratory data analyses to discover suitable  $\alpha$  levels for two eye-tracking datasets, GRE Math dataset and Online Shopping dataset. To quantitatively measure fixation identification performance by the mathematical formulation with different  $\alpha$  levels, three expert eye-tracking researchers were employed to label the subsequences randomly sampled from the eye-tracking datasets. We compared the identification results by the formulation (1.11a)–(1.11f) with different  $\alpha$  levels to the ground truth, that is, the human-labeled results. Our current finding is that the GRE Math dataset prefers  $\alpha$  at smaller levels than the Online Shopping data by observing random sampled participant eye tracking recordings. We may recommend a low  $\alpha$  value, such as 1.75, for the GRE Math recordings and a relatively high  $\alpha$  value, such as 8.5, for the Online Shopping recordings. Moreover, we may be able to generalize such recommendations for eye tracking experiments with similar tasks and static stimuli. Further verification will require additional datasets. We subsequently explore the opportunity of building a machine learning model to assign a

predictive  $\alpha$  value based on features of each input data chunk. Comparing the results from identifying fixation points in the GRE Math data chunks, reveals that running the formulation (1.11a)–(1.11f) with the  $\alpha$  levels predicted by the learning model outperforms those with fixed  $\alpha$  levels. However, due to the limited testing data, we could not draw the conclusion with statistical significance. More eye tracking data is needed for further improvement.

Future work consists of incorporating the above research into a real-time system for gaze fixation detection using inner-density. This system aims to analyze and categorize eye gaze data in near real-time, thereby establishing a basis for immediate feedback to the user.

Tobii [29] Pro software development kit (SDK) serves as the fundamental SDK to create our application for analyzing eye tracking data. It is compatible with Tobii Pro eye tracker hardware [29] that we used for collecting eye gaze datasets in previous computational experiments. Based on the SDK, we will develop our own software for realizing the FID and FID+ filter in practical use. The SDK provides raw gaze data stream with high-precision timestamps. Once the data arrives, we can save data points into a data buffer to form a data sequence. For each raw gaze point, the software could first pass it through an I-VT filter to identify whether it is a saccade point. The data sequence in the data buffer can then be separated into data chunks by saccade points. The distinct data chunks are the input of the mathematical formulation in the FID/FID+ filter. Subsequently, fixation points can be used to analyze, store or visualize user behavior and attention.

This real-time eye tracking feedback system has significant implications, and may expand the application scope of eye tracking in affective computing and accessibility. For example, if the system recognizes fixations in real-time and interprets user eye movement mode as focused attention while viewing information, the system could then provide explanatory feedback. With our advanced fixation detection algorithms, the system has the potential to improve user experience by developing innovative, personalized human-computer interactions. It may also enhance the performance of gaze-based alternative and augmentative communication (AAC) devices, which promises significant benefits to people with disabilities.

# References

- [1] Soussan Djamasbi. Eye tracking and web experience. *AIS Transactions on Human-Computer Interaction*, 6(2):37–54, 2014.
- [2] Xavier Radvay, Stéphanie Duhoux, Françoise Koenig-Supiot, and François Vital-Durand. Balance training and visual rehabilitation of age-related macular degeneration patients. *Journal of Vestibular Research*, 17(4):183–193, 2007.
- [3] Glenn C Cockerham, Eric D Weichel, James C Orcutt, Joseph F Rizzo, and Kraig S Bower. Eye and visual function in traumatic brain injury. *Journal of Rehabilitation Research and Development*, 46(6):811–818, 2009.
- [4] Michel Wedel and Rik Pieters. A review of eye-tracking research in marketing. *Review of Marketing Research*, 4:123–147, 2008.
- [5] Joseph H Goldberg, Mark J Stimson, Marion Lewenstein, Neil Scott, and Anna M Wichansky. Eye tracking in web search tasks: Design implications. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, pages 51–58. ACM, 2002.
- [6] Soussan Djamasbi, Marisa Siegel, and Tom Tullis. Generation Y, web design, and eye tracking. *International Journal of Human Computer Studies*, 66(5):307–323, 2010.
- [7] Marcus Nyström and Kenneth Holmqvist. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42(1):188–204, 2010.
- [8] 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*, pages 71–78. ACM, 2000.
- [9] Pieter Blignaut. Fixation identification: The optimum threshold for a dispersion algorithm. *Attention, Perception, & Psychophysics*, 71(4):881–895, 2009.

- [10] Mina Shojaeizadeh, Soussan Djamasbi, and Andrew C. Trapp. Density of gaze points within a fixation and information processing behavior. In *Proceedings of the 2016 Human-Computer Interaction International (HCII) Conference*, pages 1–8. Springer, 2016.
- [11] Dimitris Bertsimas and Angela King. OR Forum—An algorithmic approach to linear regression. *Operations Research*, 64(1):2–16, 2015.
- [12] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halzka Jarodzka, and Joost Van de Weijer. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press, 2011.
- [13] Oleg V. Komogortsev, Denise V. Gobert, Sampath Jayarathna, Do Hyong Koh, and Sandeep M Gowda. Standardization of automated analyses of oculomotor fixation and saccadic behaviors. *IEEE Transactions on Biomedical Engineering*, 57(11):2635–2645, 2010.
- [14] Vladimir Estivill-Castro. Why so many clustering algorithms: A position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002.
- [15] Maura Sabatos-DeVito, Sarah E Schipul, John C Bulluck, Aysenil Belger, and Grace T Baranek. Eye tracking reveals impaired attentional disengagement associated with sensory response patterns in children with autism. *Journal of Autism and Developmental Disorders*, 46(4):1319–1333, 2016.
- [16] Emilia Thorup, Pär Nyström, Gustaf Gredebäck, Sven Bölte, and Terje Falck-Ytter. Altered gaze following during live interaction in infants at risk for autism: An eye tracking study. *Molecular Autism*, 7(12):1–10, 2016.
- [17] Anneli Olsen. The Tobii I-VT fixation filter. *Copyright© Tobii Technology AB*, 2012.
- [18] Jeroen BJ Smeets and Ignace TC Hooge. Nature of variability in saccades. *Journal of Neurophysiology*, 90(1):12–20, 2003.
- [19] Evimaria Terzi. *Problems and Algorithms for Sequence Segmentations*. PhD thesis, The University of Helsinki, Finland, 2006.
- [20] Ella Bingham. Finding segmentations of sequences. In *Inductive Databases and Constraint-Based Data Mining*, pages 177–197. Springer, 2010.
- [21] Andrew Trapp, Oleg A Prokopyev, and Stanislav Busygin. Finding checkerboard patterns via fractional 0–1 programming. *Journal of Combinatorial Optimization*, 20(1):1–26, 2010.

- [22] Andrew C. Trapp and Oleg A. Prokopyev. Solving the order-preserving submatrix problem via integer programming. *INFORMS Journal on Computing*, 22(3):387–400, 2010.
- [23] Onur Seref, Ya-Ju Fan, and Wanpracha Art Chaovalitwongse. Mathematical programming formulations and algorithms for discrete  $k$ -median clustering of time-series data. *INFORMS Journal on Computing*, 26(1):160–172, 2013.
- [24] Beibin Li, Quan Wang, Erin Barney, Logan Hart, Carla Wall, Katarzyna Chawarska, Irati Saez de Urabain, Timothy J Smith, and Frederick Shic. Modified DBSCAN algorithm on oculomotor fixation identification. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 337–338. ACM, 2016.
- [25] M.R. Rao. Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335):622–626, 1971.
- [26] Tai-Hsi Wu. A note on a global approach for general 0–1 fractional programming. *European Journal of Operational Research*, 101(1):220–223, 1997.
- [27] Alizadeh Farid and Goldfarb Donald. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.
- [28] Purvi Shah, Minal Goyal, and Daiyang Hu. Role of expiration dates in grocery shopping behavior: An eye tracking perspective. In *Proceedings of the Twenty-Second Americas Conference on Information Systems (AMCIS)*, pages 1–5. Association for Information Systems, 2016.
- [29] Tobii. Tobii technology. <http://www.tobii.com>, 2019. Accessed: 2019-04-02.
- [30] Gurobi Optimization Inc. Gurobi Optimizer 7.5.1 Reference Manual. <http://www.gurobi.com>, 2018.
- [31] MathWorks Inc. MATLAB Users Guide, 2016.
- [32] Jarkko Salojärvi, Kai Puolamäki, Jaana Simola, Lauri Kovanen, Ilpo Kojo, and Samuel Kaski. Inferring relevance from eye movements: Feature extraction. In *NIPS Workshop*, pages 45–67, 2005.
- [33] Andy Mitchell. The ESRI guide to GIS analysis II: spatial measurements and statistics ESRI Press. *Redlands CA*, 2005.

- [34] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods*, pages 1–22, 2016.
- [35] Kenneth Holmqvist, Marcus Nyström, and Fiona Mulvey. Eye tracker data quality: What it is and how to measure it. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 45–52. ACM, 2012.
- [36] Marcus Nyström, Richard Andersson, Kenneth Holmqvist, and Joost Van De Weijer. The influence of calibration method and eye physiology on eyetracking data quality. *Behavior Research Methods*, 45(1):272–288, 2013.
- [37] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1118–1130. ACM, 2017.
- [38] Pieter Blignaut and Daniël Wium. Eye-tracking data quality as affected by ethnicity and experimental design. *Behavior Research Methods*, 46(1):67–80, 2014.
- [39] Oleg V Komogortsev and Javed I Khan. Eye movement prediction by oculomotor plant Kalman filter with brainstem control. *Journal of Control Theory and Applications*, 7(1):14–22, 2009.
- [40] Michiel Smid. Finding  $K$  points with a smallest enclosing square. In *Report MPI-92-152, Max-Planck-Institut für Informatik*, 1993.
- [41] Raimondas Zemblys. Eye-movement event detection meets machine learning. *Biomedical Engineering*, 20(1), 2016.
- [42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [43] Anjith George and Aurobinda Routray. A score level fusion method for eye movement biometrics. *Pattern Recognition Letters*, 82:207–215, 2016.
- [44] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

- [45] Art B Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7):59–72, 2007.