Highlighting the Challenges of Blinks in Eye Tracking for Interactive Systems

**Jesse W. Grootjen**  
LMU Munich  
Munich, Germany  
Munich Center for Machine Learning (MCML)  
Munich, Germany  
jesse.grootjen@ifi.lmu.de

**Henrike Weingärtner**  
LMU Munich  
Munich, Germany  
henrike.weingaertner@ifi.lmu.de

**Sven Mayer**  
LMU Munich  
Munich, Germany  
info@sven-mayer.com

---

**ABSTRACT**

Eye tracking is the basis for many intelligent systems to predict user actions. A core challenge with eye-tracking data is that it inherently suffers from missing data due to blinks. Approaches such as intent prediction and user state recognition process gaze data using neural networks; however, they often have difficulty handling missing information. In an effort to understand how prior work dealt with missing data, we found that researchers often simply ignore missing data or adopt use-case-specific approaches, such as artificially filling in missing data. This inconsistency in handling missing data in eye tracking hinders the development of effective intelligent systems for predicting user actions and limits reproducibility. Furthermore, this can even lead to incorrect results. Thus, this lack of standardization calls for investigating possible solutions to improve the consistency and effectiveness of processing eye-tracking data for user action prediction.

---

**CCS CONCEPTS**

- Human-centered computing → Human computer interaction (HCI); Interactive systems and tools.

**KEYWORDS**

Human-computer interaction, eye tracking, blink detection, interactive systems

---

**1 INTRODUCTION**

Eye tracking has become an essential component of intelligent systems, providing valuable information for predicting user actions. Despite its importance, eye-tracking data, mainly when acquired in the wild, is inherently plagued by missing information due to blinks, making it impossible for today’s neural network approaches to predict user intentions accurately. Therefore, today’s methods all require additional pre-processing steps for the handling of missing data. We can find a set of established methods for blink detection. When looking into how to handle the blinks, literature is less standardized as researchers often employ use-case-specific approaches to deal with these gaps in data. Thus, we argue there is currently
no standardized approach to handling missing eye-tracking data for intelligent systems.

Past research has attempted to address this challenge by adopting use-case-specific approaches, such as removing parts of data that include missing data or artificially filling it in, e.g., [Stein et al. 2022; Wang et al. 2021]. However, this lack of consistency in handling missing data presents a significant challenge, and a more standardized solution is needed to ensure the accuracy and effectiveness of intelligent systems for user action prediction. This issue highlights the importance of developing a comprehensive and consistent approach to processing eye-tracking data.

We searched for relevant papers to understand how researchers have dealt with missing data in eye tracking to combat this issue. In this work, we highlight several different use-case-specific approaches and demonstrate the need to standardize data processing techniques. We also speculate about the impact a standardized way of pre-processing could have on interactive systems. Improper handling of blinks and missing data can result in inaccurate and inconsistent predictions, as well as distorted data analysis. Furthermore, the lack of a standardized approach to data processing presents a significant challenge to the reproducibility and comparability of results across different studies. Therefore, it is crucial to establish a comprehensive and consistent approach to pre-processing eye-tracking data to ensure the reliability and validity of intelligent systems for user action prediction.

2 RELATED WORK

First, we provide a short overview of the reasons for blinks and how blinks are used in interactive systems for human-computer interaction (HCI). Next, we provide insight into different ways of blink detection. For the final part of our related work, we provide more cases of the use of eye tracking in interactive systems.

2.1 Reasons for Blinks

Blinks are a natural and essential part of human visual perception and are defined as the rapid closure and reopening of the eyelids. They typically last between 100 and 400 milliseconds and occur spontaneously, approximately every 2 to 10 seconds, depending on various factors, such as age and environment [Fatt and Weissman 2013]. While blinks are crucial for maintaining ocular surface health and protecting the eyes from damage caused by airborne particles [Stern et al. 1994], excessive light [Patel et al. 1991], and other environmental factors, they also pose a challenge for eye-tracking systems.

Blink data can also be used to assess cognitive and perceptual processes in a variety of domains, such as neuroscience, psychology, and ophthalmology. As an example of this, researchers have used blink frequency to examine attentional processing, including selective attention, inhibition, and cognitive load [Tsai et al. 2007; Wolkoff et al. 2005]. In ophthalmology, blink rate has been used as an index of corneal sensitivity and tear film quality. Furthermore, blink dynamics can be used to diagnose and monitor various neurological and neuromuscular disorders, such as Parkinson’s disease [Bek et al. 2020], myasthenia gravis [Nguyen et al. 2022], and Tourette’s syndrome [Shaikh et al. 2017]. While accurately capturing blink data is critical for drawing meaningful inferences and diagnosing these conditions, blinks are influenced by many factors, which can heavily impact the accuracy of these systems.

2.2 Impacting Factors on Blinks

Various factors can impact how blinks manifest. However, blink frequency and blink duration are the two variables that characterize blinks most predominately.

2.2.1 Blink Frequency. Various factors, including age and gender [Zhan et al. 2020], as well as environmental conditions, and cognitive load, influence blink frequency. For example, younger children blink much less frequently than adults, with blink rates increasing gradually from infancy through early childhood, after which it decreases again with age [Marandi and Gazerani 2019; Stern et al. 1994]. Women tend to blink more frequently than men. Environmental factors such as lighting conditions, air quality, and screen usage can also impact blink frequency. Prolonged screen use, in particular, can cause a decrease in blink frequency [Patel et al. 1991], leading to symptoms such as dry eyes and eye strain. Furthermore, cognitive load, such as engaging in a demanding task, can cause a reduction in blink frequency [Tsai et al. 2007; Wolkoff et al. 2005]. These factors can vary across individuals and situations, highlighting the importance of understanding the context in which blinks occur to model and interpret eye-tracking data accurately.

2.2.2 Blink Duration. Blink duration is another important factor to consider in eye-tracking research. Blink duration is affected by many of the same factors that influence blink frequency, including age, gender, environmental conditions, and cognitive load. On average, blink durations range from 100 to 400 milliseconds [Fatt and Weissman 2013], but they can vary widely. For example, blink duration tends to be longer in older adults and shorter in younger adults. Environmental factors such as lighting and screen usage can also impact blink duration. Prolonged screen use can cause a decrease in blink frequency and an increase in blink duration, which can contribute to symptoms such as dry eyes and eye strain. Furthermore, cognitive load can influence blink duration, with longer blinks being associated with more demanding tasks. Understanding these factors is crucial for accurately interpreting blink data and improving the performance of eye-tracking systems.

2.2.3 Additional Factors. The four factors below are additional characteristics of blinks. Inter-blink interval: The time between blinks can also be influenced by various factors, such as mental workload [Wilson and Russell 2003], fatigue [Ousler et al. 2002], and emotional states [Keil et al. 2006]. Blink asymmetry: Studies have shown that the timing and duration of blinks can differ between the left and right eyes, which may be related to differences in motoneuron excitability [Kassem and Evinger 2006]. Blink rate variability: While the average blink rate is an important metric, the variability in blink rates can also provide valuable information about cognitive [Paprocki and Lenskiy 2017] and emotional states [Godin et al. 2015]. Blink patterns: In addition to duration and frequency, the pattern of blinks (e.g., regular, irregular, spontaneous) can also provide insights into cognitive processes [Jongkees and Colzato 2016] and attentional states [Lawson et al. 1998].
2.3 Blink Detection Methods

Various methods have been proposed for detecting blinks in eye-tracking data, ranging from intrusive to non-intrusive techniques [Holmqvist et al. 2011]. Intrusive methods include using electrodes placed around the eyes, such as electrooculography (EOG) [Tamba et al. 2014]. However, these methods can be uncomfortable and may disrupt their natural behavior. In contrast, non-intrusive methods use cameras to track eye movements, either with or without additional illuminators. Examples of non-intrusive methods include the use of infrared cameras, which can detect pupil movement even in low-light conditions, or using visible light cameras to track facial features such as the eyes and eyebrows [Fejtová et al. 2004]. In the following section, we will discuss some of the most widely used non-intrusive blink detection approaches in eye-tracking systems.

Some eye tracking systems, such as the EyeLink 1000 Plus1 (SR Research Ltd., Ottawa, ON, Canada), have built-in blink detection algorithms that are designed to identify blinks automatically. Their algorithm offers a blink detection method whereby a blink is defined as periods of eye-position data in which the pupil size is notably small or when the pupil in the camera image is missing or severely distorted due to eyelid occlusion. The parser senses partial occlusions of the pupil preceding and following a blink, which is then marked as a saccade. The parser recommends discarding fixations lasting less than 100 milliseconds before and after a blink to minimize the effect of blink-related artifacts. These procedures are detailed in their manual.

Another built-in blink detection method comes from the BeGaze parser2, developed by SensoMotoric Instruments Gmbh in Teltow, Germany. It incorporates a pre-built blink detector that identifies blink events as a particular type of fixation where the pupil diameter is either zero or outside a dynamically computed valid pupil range. The blink event is expanded to include this period, which is determined based on changes in pupil diameter to account for the transition period between valid gaze data and a blink. If the pupil diameter exceeds an internal threshold value, the system considers this period to be part of the blink. The processing discards blink events that are shorter than 70 milliseconds.

Compared to built-in methods, custom blink detection models have been proposed in recent years as an alternative. Various publications have demonstrated the effectiveness of these models, e.g., Al-Hindawi et al. [2022]; Appel et al. [2016]; Królok and Strumillo [2012]. One example is the blink detection method proposed by Al-Gawwam and Benaissa [2017], which uses facial features from a video sequence instead of specifically looking at the eyes, making it robust against various illumination and facial expressions. Another example is the model developed by Hu et al. [2020], which uses pictures from eyes to classify for a blink or not using AdaBoost and ANN. Although these methods offer novel approaches to blink detection, they are all RGB-camera-based and cannot be applied to data collected with existing eye trackers.

2.4 Interactive Systems

Gaze-based interaction techniques have been widely used in various interactive systems, such as target selection [Li et al. 2021], navigation [Giannopoulos et al. 2015], and input via gaze gestures [Drewes and Schmidt 2007]. For instance, intentional blinks have been shown to be a feasible technique for menu navigation in high-risk scenarios like air traffic control [Traoré and Hurter 2016]. Similarly, dwell time has been a popular input method for gaze-based interaction, allowing users to select items or navigate menus by fixating on the target for a specific duration. Compared to traditional input methods like a mouse, eye tracking-based input with dwell time has been shown to perform better in selecting visual targets [Alonso et al. 2013; Komogortsev et al. 2009].

Predicting gaze behavior in interactive systems is another area of interest, as it can enable the evaluation and improvement of these systems without requiring a human user. Saliency maps and task-specific models like EZ Reader have been used to predict gaze behavior in various settings, such as short videos and virtual reality (VR) environments [Feng et al. 2013]. Predicting gaze in short videos can be used for interactive media applications like customized advertisements, while gaze prediction can be used to pre-render scenes in VR to improve the user experience [Xu et al. 2018] or to support next action prediction [Zhang et al. 2022].

3 LITERATURE SEARCH

We identified several ways of blink detection methods applied in related work, which we found through our search. While for the SMI and EyeLink trackers, there are exciting parsers available, most of the identified literature simply reports on the missing data being the detector for an indication of a blink. For example, Li et al. [2020] reported on missing data using the EyeLink and others [Bhattacharya et al. 2020; Keshava et al. 2020; Koskinen and Bednarik 2020] report on missing data from SMI eye tracker. As such, we identified that the blink detection methods do not leverage the available parsers, and improving this has the potential for a positive impact on research quality.

Although various solutions for detecting blinks and handling missing data have been proposed, there are no readily available solutions integrated into the current eye-tracking pipeline. Since eye-tracking data is a time-series, traditional interpolation methods, such as replacing missing data with the mean or median, are not appropriate for handling missing data caused by blinks. Therefore, the majority of previous studies identified with our search have resorted to removing the data containing blinks. Nevertheless, alternative methods for handling missing data caused by blinks have been explored in the literature, such as interpolation and imputation. Despite being less commonly used, these approaches have the advantage of preserving the applicability of eye-tracking data in interactive systems.

3.1 Remove

Most of the identified literature on blink detection and handling involves removing the data that contains blinks. However, some studies have proposed criteria for retaining the blinks in the data. For instance, Ishii et al. [2013]; Nakano and Ishii [2010] remove data when blinks last longer than 200 milliseconds, while they merge data before and after the blink if it is shorter than 200 milliseconds. Similarly, Bixler and D’Mello [2021] allow for up to 20% of the data
3.2 Interpolate

Interpolation is a frequently employed technique for replacing the gaps in the data introduced by blinks, as reported in the identified literature. The work utilized various forms of interpolation, such as linear, polynomial, spline, and cubic spline. For instance, Kinnunen et al. [2010] applied linear interpolation by assuming the continuity of the data for the parts with missing data and independently applying interpolation to all coordinate time series. Another example is Wang et al. [2021], where the authors set missing data when the pupil size was outside a range and then applied spline interpolation to correct the missing values.

3.2.1 Linear Interpolation. Linear interpolation is, of the identified literature, the most commonly used method to deal with missing data in eye-tracking studies after removing the data. This technique assumes that the underlying signal is a linear function and fills in the missing data by connecting the available data points with straight lines. While linear interpolation is a simple and efficient method, it can produce artifacts in the interpolated data and can introduce errors in the analysis of the eye-tracking data.

An example of the use of the linear interpolation method is in Bafna et al. [2020]. In this work, the authors removed 200 ms before and after a blink using the Tobii 4C Eye Tracker during a typing task. Another example of this is Saluja et al. [2019], here the authors also used linear interpolation. However, this was during a reading task and without removing additional data before and after the blink.

3.2.2 Spline Interpolation. Spline interpolation is another method we have seen in the identified literature more than once. Unlike linear interpolation, which assumes a constant rate of change between data points, spline interpolation uses piecewise polynomials to interpolate data, resulting in a smoother fit. Cubic spline interpolation is a specific type of spline interpolation that uses cubic polynomials to interpolate between data points. Spline interpolation has been used in eye-tracking studies to fill in missing data due to blinks, head movements, or other sources of noise.

For example, Wang et al. [2021] used spline interpolation to correct for missing data when the pupil samples were outside of a pre-determined range during a driving simulator experiment. For cubic spline interpolation, we identified Morales et al. [2018]. Here the authors used the EyeLink and its online parser software to identify blinks, after which they applied cubic splines to remove blink data.

3.2.3 Polynomial Interpolation. Polynomial interpolation is another method that can be used for infilling missing values in eye-tracking data. In this method, a polynomial function is used to fit the known data points, and the polynomial is then used to estimate the values of the missing data points. An advantage of polynomial interpolation is that it is easy to implement, and it can be used to estimate missing data points with a high degree of accuracy. However, polynomial interpolation is also sensitive to outliers in the data, and it can produce unrealistic estimates if the data contains extreme values or abrupt changes in eye movement patterns. An example of the use of polynomial interpolation we identified in our search is the work from Keshava et al. [2020], here the authors employ the interpolation method to the eye movements from a VR task.

3.3 Other

Finally, we identified a range of alternative methods to deal with missing data caused by blinks, including imputation, WEKA, aggregating, and winsorizing. For instance, Li et al. [2020] employed the unsupervised Expectation-Maximization algorithm to impute missing values in their eye-tracking data. Likewise, Cole et al. [2015] filled in the missing data using observed session transition probabilities. Despite the various approaches identified, there is no consensus in the literature on which method is the most appropriate for dealing with blinks. While most reviewed studies remove the affected data, this often leads to a significant loss of data.

4 DISCUSSION

In this work, we identified several ways past work has employed to identify blinks and algorithms used to infill the missing data and found that these are not consistent throughout the existing literature. In the following, we will showcase the challenges of the different interpolation methods and the impact of incorrect cut-off times can have on the interpolation.

**Linear interpolation.** Linear interpolation is a simple and commonly used infilling method for missing data in the identified literature. The method involves drawing a straight line between pairs of data points that flank a gap and then interpolating data points along the lines to fill in the missing values. From the identified infilling methods used, linear interpolation was the most common one, which we attribute to its simplicity and low computational cost. However, as we visualize in Figure 2, the method assumes that the data points before and after the gap are linearly related.

**Figure 2:** An example of linear interpolation over a blink during a reading experiment. Here, the black dots represent fixations, the black line eye movements between the fixations and the red lines represent the interpolation. In (a), we visualize the interpolation without removing any of the information pre or post-blink. In (b), we visualize the interpolation with the removal of information pre and post-blink, however, not enough of the artifacts have been removed. In (c), we visualize the correct removal of artifacts pre and post-blink. Finally, in (c), we showcase the removal of too much of the data pre and post-blink.
While this method is not as popular in the identified literature as linear interpolation, which is not always the case. Additionally, the method can produce abrupt changes in velocity and acceleration, as most predominantly visualized in Figure 2a and Figure 2b. As such, when using linear interpolation, a best practice would be to visually inspect each individual blink to see where artifacts of blink behavior are introduced into the data. Removing these artifacts will allow for the best possible interpolation between data gaps, as visualized in Figure 2c. However, overdoing the cut-off before and after the artifacts will introduce new noise into the data, as visualized in Figure 2d.

**Spline interpolation.** Spline interpolation is a more sophisticated way of interpolating compared to linear interpolation. This method fits a piecewise polynomial function to the data points where the resulting function is smooth and passes through all data points, including those with missing data. It allows for considering the velocity and direction of the data points pre and post-missing data to ensure the curve will not abruptly change velocity or acceleration. While this method is not as popular in the identified literature as linear interpolation, it could provide a more smooth and more accurate interpolation compared to linear interpolation. However, spline interpolation requires more computational resources than linear interpolation. Still, this interpolation method struggles with the artifacts introduced pre and post-blink as visualized in Figure 3a and Figure 3b. When visualizing the exact cut-off where the artifacts from blinks start in Figure 3c, the infill method results in a smooth curve taking the velocity and acceleration into account before and after the gap. However, when cutting off too much of the data as visualized in Figure 3d, we can see that the infilling method misses quite a few of the words that would otherwise be in the path of the eye path.

**Polynomial interpolation.** Polynomial interpolation is the last method we identified in multiple papers throughout the identified literature. It fits a single polynomial function to all the data points, which is different from spline interpolation, which divides all data points into segments. The resulting function is smoother than linear interpolation but not as smooth as spline interpolation.

In conclusion, interpolation methods provide a powerful tool for handling missing data in eye-tracking studies. We argue that interpolation methods should be preferred over removing data that contains blinks, as this results in a large portion of the data becoming unusable. Linear interpolation is the most commonly used interpolation method in the identified work and provides an accurate estimate of the eye movements when the artifacts introduced by blinks are correctly cut-off. However, this method can produce abrupt changes in velocity and acceleration when the artifacts are either ignored or improperly dealt with. Spline interpolation is a more sophisticated method that provides a smooth and accurate estimate but requires more computational resources compared to linear and polynomial interpolation. While this method is less sensitive to abrupt changes in velocity and acceleration, appropriately handling the artifacts introduced before and after a blink will yield the best result. Lastly, polynomial interpolation is simple and computationally efficient, similar to linear interpolation. However, the accuracy of this method heavily relies on the linear nature of the eye movements before and after the blink, as such appropriately dealing with the artifacts is, especially with this method, hugely important.

The choice of the interpolation method will depend on the specific characteristics of the eye movements and, thus, indirectly depend on the nature of the experiment where the data is collected. In all cases of interpolating, the challenge is correctly identifying...
and removing the artifacts introduced pre and post-blink. In the literature identified, we only found one case where the authors removed data pre, and post-blink and another where the authors relied on the included parsing algorithm. The remaining works only mention that blinks are identified as missing data and do not specify any additional data being removed either pre or post-blink.

ACKNOWLEDGMENTS

Sven Mayer was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project-ID 251654672- TRR 161.

REFERENCES


1719970.1719990


