Rule-based learning for eye movement type detection

Wolfgang Fuhl
University of Tübingen, Perception Engineering
wolfgang.fuhl@uni-tuebingen.de

Nora Castner
University of Tübingen, Perception Engineering
castnern@informatik.uni-tuebingen.de

Enkelejda Kasneci
University of Tübingen, Perception Engineering
enkelejda.kasneci@uni-tuebingen.de

ABSTRACT

Eye movements hold information about human perception, intention, and cognitive state. Various algorithms have been proposed to identify and distinguish eye movements, particularly fixations, saccades, and smooth pursuits. A major drawback of existing algorithms is that they rely on accurate and constant sampling rates, error free recordings, and impend straightforward adaptation to new movements, such as microsaccades, since they are designed for certain eye movement detection. We propose a novel rule-based machine learning approach to create detectors on annotated or simulated data. It is capable of learning diverse types of eye movements as well as automatically detecting pupil detection errors in the raw gaze data. Additionally, our approach is capable of using any sampling rate, even with fluctuations. Our approach consists of learning several interdependent thresholds and previous type classifications and combines them into sets of detectors automatically. We evaluated our approach against the state-of-the-art algorithms on publicly available datasets. Our approach is integrated in the newest version of EyeTrace which can be downloaded at *http://www.ti.uni-tuebingen.de/Eyetrace.1751.0.html*.

CCS CONCEPTS

• **Computing methodologies → Supervised learning**; **Rule learning**; *Feature selection*; *Cross-validation*;

KEYWORDS

Eye tracking, eye movements, machine learning, saccade, fixation, smooth pursuit, post saccadic movements

ACM Reference Format:
Wolfgang Fuhl, Nora Castner, and Enkelejda Kasneci. 2018. Rule-based learning for eye movement type detection. In *Workshop on Modeling Cognitive Processes from Multimodal Data (MCPMD'18), October 16, 2018, Boulder, CO, USA,* Felix Putze, University of Bremen, Jutta Hild, Fraunhofer IOSB, Enkelejda Kasneci, University of TÃijbingen, Akane Sano, MIT Media Lab/Cornell University, Erin Solovey, Drexel University, and Tanja Schultz, University of Bremen (Eds.). ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3279810.3279844

MCPMD'18 , October 16, 2018, Boulder, CO, USA

ACM ISBN 978-1-4503-6072-2/18/10…$15.00
https://doi.org/10.1145/3279810.3279844

1 INTRODUCTION

Eye movements are a valuable resource to extract a diversity of information about a subject: Mainly, intentions and cognitive states [2, 17], such as the workload [21] and attention of a person. This information allows systems to warn humans if they are no longer capable of proper task performance like driving [12] or surgeries [4]. Additionally, eye movements aid in detecting eye diseases [20].

Initially, these detections and classifications begin with the extraction of the eye movements. This process consists of separating the raw gaze signal into segments which belong to different types of eye movements. These types hold information of physical constraints of the subject. For example, visual perception during a saccade is severely limited [14, 25]. Another issue with saccades is that the cornea is deformed during this type of eye movement and influences the shape of the pupil captured in video-based eye tracking [11, 22, 29], and therefore, influences the measured velocity. However, for smooth pursuit, where the subject follows a moving object, the perception works without restrictions [24]. Blinking is an eye movement which keeps the eye moist and protects it against particles. Since this is not a movement of the eyeball, the perception is limited in the process of closing and opening [34]. The frequency and duration of this type of movement also hold valuable information about the subject's cognitive state but is out of the scope of this paper [28]. In modern high-speed eye trackers, post-saccadic oscillations are a novel type of eye movement, which most of the state-of-the-art algorithms do not detect. It is a saccadic movement that corrects for the overshoot of the preliminary saccade [23]. In contrast to the saccade, the subject can perceive but with distortions [29].

In order to effectively detect eye movements, extraction has to be accurate and robust against noise. Modern algorithms apply different dispersion, velocity, or acceleration thresholds and validate the detected eye movement type based on the duration as well as dispersion. In [1], it was found that all approaches work unsatisfactory. Additionally, all the approaches need time to adjust the parameters, which have to be adjusted between subjects and experiments. There are several reasons why these algorithms work inadequately. The first is that all rely on the error detection of the used eye-tracking device for recording. Since this is not true for eye tracking in the wild, most methods fail because of invalid eye positions in the raw gaze data. This reliance also has an impact on the preprocessing of data, since parameters for smoothing or the method to compute eye movement velocities has to be changed. Another issue is that video cameras that are used to record the eyes in modern eye trackers do not have a stable recording rate. It can change slightly or, based on the setup, frames can be dropped because the hardware does not have enough resources available to store or process the next frame. Even more reasons can be found in [3, 5].

For the parameters of eye movement detection algorithms, different thresholds are proposed in the literature [10]. It is difficult to transfer them to eye tracking in real environments because they are determined under laboratory conditions and need accurate and noise-free recordings. As previously mentioned, not all noise is annotated by the eye tracker and the sampling rate varies, especially if there are reflections on glasses or the lighting conditions change. For these cases, the detection algorithm cannot track the pupil anymore and has to start a re-detection that consumes more resources. Therefore, the detection of eye movements is still a challenging and important task.

The choice of the algorithm and the determination of the parameters are currently up to the researcher. Generally, the tedious and difficult task of using an algorithm is to adjust its parameters. The researcher can only use literature values or manually annotate the data to check the selected parameters. Unfortunately, this process theoretically has to be repeated several times because the quality of the data often varies between subjects and between different experiments. In the case of new eye movements with no available algorithm, the researcher is on his own and there is no other choice but to annotate them manually. The proposed approach is based on machine learning where errors in the data and also new eye movements can be trained and recognized. The simple construction of our detectors and the easy understanding of the rules can also help the researcher extract characteristic properties of new eye movements.

2 RELATED WORK

The most prominent fixation and saccade detection algorithm is Identification by Dispersion-Threshold (IDT) [26]. It uses the data reduction proposed in [35]. The algorithm uses two thresholds, one is for the maximum fixation dispersion and the other for the minimum fixation duration. Another algorithm that is simple to implement is the Identification by Velocity Threshold (IVT) [26], where each sample below a chosen velocity threshold is classified as fixation and above as saccade. It is mostly applicable for high-speed recordings. Based on the IVT algorithm, a self-adaptive approach was proposed in [6, 7], where it was developed to detect microsaccades. In that approach, the velocity threshold is automatically adapted to the noise level in the eye-tracking data. An algorithm specially designed to cope with noisy data is the Identification by Kalman Filter (IKF) algorithm [16]. It uses the Kalman filter to predict the next sample value based on previous values. Therefore, it interpolates the data in an online fashion. For classification, two thresholds are used: one for the predicted value (velocity or distance) and one for the minimum fixation duration. Similar to this algorithm, an implementation using the χ^2-test instead of the Kalman filter was proposed in [15]. In [33], the Covariance Dispersion Algorithm (CDT) was proposed. It is an improvement of the F-tests dispersion algorithm (FDT) [32]. The F-test measures if two data samples belong to the same class and, due to the assumption that the data follows normal distributions, it is sensitive to noise. An improvement by the covariance matrix is introduced to cope with this problem. The algorithm needs three thresholds, one for the variance, one for the covariance, and a third threshold for the minimum duration. The identification by a Minimal Spanning Tree (IMST) [15] creates a tree upon the data, where the samples represent the leafs. The goal is to select all samples with a minimum of branches given a connected graph (the data). Hidden Markov Models (HMM) have been proposed in [13, 15, 27, 30] to separate fixations from saccades and even to detect smooth pursuits. The HMM consists of at least two states (fixation and saccade). For each new velocity sample, the model decides whether it belongs to the current state (classification) or if a state transition has occurred. After each sample, the model is updated to adapt to the data. The first algorithm able to detect post saccadic movements was proposed in [23]. Based on the noise in the data, the algorithm also adapts its velocity thresholds. The Binocular-Individual Threshold (BIT) algorithm [31] was also designed to detect small saccades in noisy data. Therefore, it applies its thresholds to the data of both eyes, following the ideas that both eyes have to perform the same movement. This algorithm also adapts its thresholds automatically. An algorithm detecting fixations, saccades, post saccadic movement, and smooth pursuits was proposed in [19]. This algorithm adapts the parameters automatically and is the first method capable of detecting all these eye movements at the same time. For high-speed eye-tracking data, an algorithm for fixation, saccade, and smooth pursuit detection was proposed in [18]. The algorithm uses three stages to classify the data, starting with a preliminary segmentation and then evaluating each segment again, followed by the final classification.

3 METHOD

In state-of-the-art algorithms there are thresholds for upper and lower limits as well as for ranges that need to de determined. The main disadvantage is that those thresholds are difficult to adjust to new data, where the sampling rate is not constant or no time information is given [1]. Another issue with those thresholds is that, for some data, they work very well while for more noisy data they do not work at all or need intensive preprocessing (such as smoothing filters and outliers detection). This preprocessing also need parameters to be adjusted like the smoothing factor, the window size, and the influence of each position in the window. Our idea is to use the traditional thresholding and smoothing approach, but to adapt the algorithm to the data automatically and also select the best parameters for the smoothing. The algorithm creation and the data smoothing can be selected for offline or online usage of the detector.

(1) Generate data Manipulation
  (a) Repeat until undetected types<ϵ_1
    (i) Repeat until error<ϵ_2 or iterations==max iterations
    (ii) Select best rule for each Type
    (iii) Add rule to current Detector per Type
  (b) Evaluate Detectors
  (c) Add best Detector to Classificator
  (d) Detect types on training set with Classificator
(2) Classificator training finished

The enumeration 2 is the work flow of the training algorithm. It first starts by generating all possible data Manipulations. In our implementation, these are the velocities computed in different time

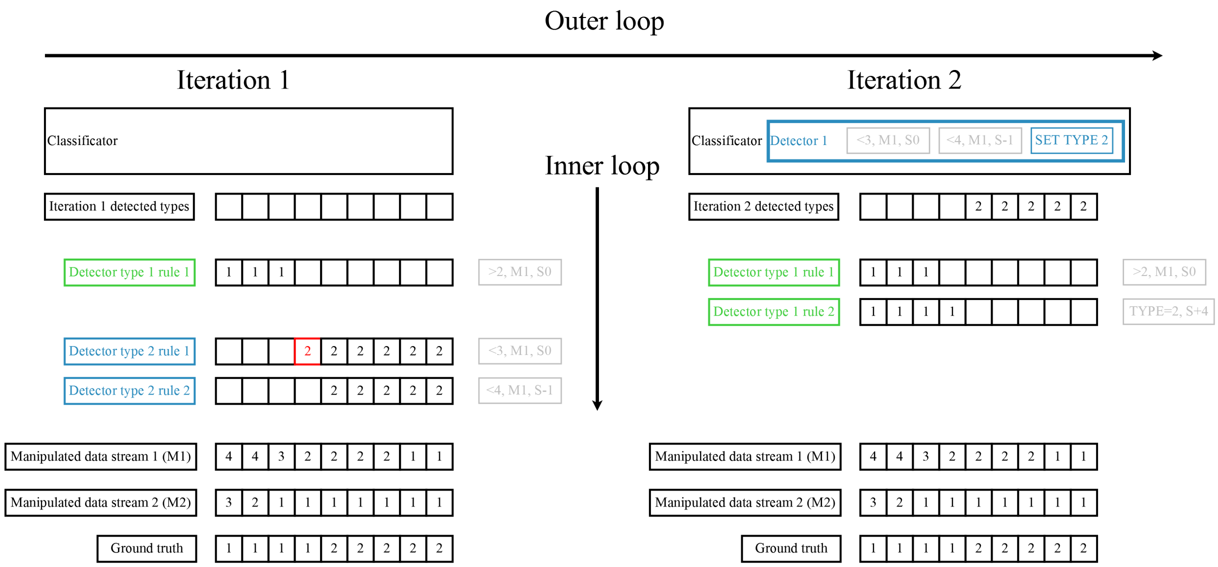

Figure 1: Exemplary training of a detector. The outer loop (left to right) collects detectors and provides the current classification results. The inner loop selects combinations of rules until the breaking conditions are met.

windows. We also provide parameterizable functions for the acceleration, angles, and averaging, where x, y or both values can be taken into account. For the evaluation, we only used the window based velocity function.

The second loop (i) generates Detectors for each eye movement type (T). It starts by evaluating each Manipulated data stream and selects the best overall rule on them. The possible Rules to learn are $M_i(S) < TH$, $M_i(S) > TH$ and $M_i(S) == TYPE$. M_i is the data Manipulation with index i. S is the displacement (the Shift) to the inspected position, TH, which is a value for comparison and $TYPE$ is a detected eye movement type (not the annotated one).

In the first iteration we select the Rule that maximizes the distance between correct(true positive TP) and misclassified (false positive FP) types. This is done because one rule is not capable of being a robust detector, but sets the limit for the overall capability of the detector to classify the data correctly. Therefore, it is allowed to misclassify data if it classifies more data correctly.

$$\underset{i,j,k}{\text{argmax}}\; TP(D(M_i, S_j, R_k)) - FP(D(M_i, S_j, R_k)) \tag{1}$$

Equation 1 formulates the maximization rule where i is the selected Manipulated data stream, j the displacement to the inspected position, and k the selected rule.

Each consecutive iteration adds a Rule which minimizes the error of the Detector because the first iteration was allowed to misclassify.

$$\underset{i,j,k}{\text{argmin}}\; FP(D^t)|TP(D^t) > TP(D^{t-1}) * (1 - \frac{ED}{2}) \tag{2}$$

Equation 1 formulates the minimization where t is the iteration and $ED = \frac{FP(D^t)}{FP(D^{t-1})}$ is a factor computed based on the reduction of the error. Meaning, if the misclassification was reduced by 10%, the detector would need to still classify 95% of the previous result correctly.

This process is repeated until the misclassification falls below a predefined ϵ_1 or the maximal number of iterations is reached. In our implementation, we set $\epsilon_1 = 0.001$ and the maximum of iterations to five.

After all detectors are created, we select the Detector with the lowest misclassification rate. If there exist multiple, the one with the highest detection rate is selected. This detector is added to the Classificator. Afterward, this Classificator is used to detect and store eye movement types in the training data. This is necessary for rule three ($M_i(S) == TYPE$) to be applicable and for the termination criterion (a). Here, the idea is to apply the learned detectors consecutively so that subsequent ones can use the results of previous detectors. Meaning, they are not allowed to override results from previous detectors. In figure 1, an example of the training procedure is shown. In the first iteration, detectors for type one and two are searched for since both still exist training data. In the inner loop, rules which fulfill equation 1 are searched for. Here, the rule where a previous set type is searched ($M_i(S) == TYPE$) cannot apply because no type was set so far. For the type one detector, one rule is enough and the result cannot be improved further since the false positive rate is zero. The type two detector, in contrast, produces one misclassification with the first selected rule but has a true positive rate of one hundred percent. With equation 2, this error is minimized to zero by selecting an additional rule. Since both detectors now have the same false positive rate, the one with the highest true positive rate is selected, which is the detector for type two. This finishes the first iteration and starts the detection and next iteration of the outer loop. In this iteration, all eye movements of type two are already classified, which means that no detector has to be trained for this. The inner loop selects two rules, where the second rule uses information from previous detection results. In this case, at position +4 (four timestamps in the future), it means that it is not an online classifier. For an online classificator, positive shifts have to be forbidden, which can be specified at the beginning of the training.

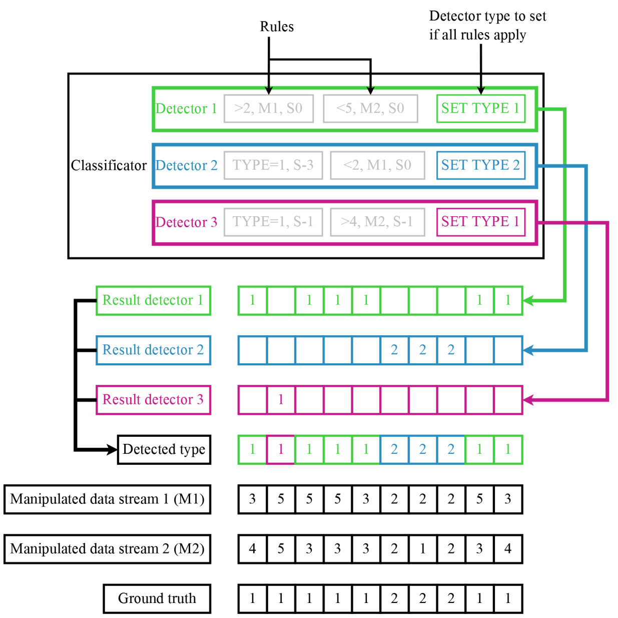

Figure 2: The consecutive detection of three detectors and two manipulated data streams. First detector one is applied and afterwards two and three. Detector two and three can use the information of the previous detections.

The final classificator consists of multiple detectors for different types. In figure 2 an detection example is shown. The first classification is performed by detector one (green) which detects nearly all type one eye movements. Afterward, detector two (blue) is applied, which also relies on the results of the first detector (first rule). This detector detects all type two eye movements. The final detector three (pink) sets the last type one eye movement type and also relies on the detections from detector one. Since all detectors of the classificator rely only on the current or past information, it is an online classificator.

4 EVALUATION

For comparison, we used the state-of-the-art algorithms from [27] (IBDT), [23] (EV), [9] (I2MC), and [19] (LS). We evaluated all algorithms on the publicly available data set from [27] (DS-IBDT) and [1] (DS-AND). For the data set from [1], we only used the annotations from the annotator *MN*. Since we are interested in the evaluation of real data, we did not remove noise nor unannotated data. Each recording was given to the algorithm without modification. For the final analysis of the detection rates, we only considered the annotated data points. Meaning the detected eye movement type for unannotated data points was ignored and has no influence on the resulting statistics.

For our proposed approach, we considered two experiments one with only three example files given for training and one where it was trained on the entire remaining data (without files from the same subject) together with simulated data using the generator from [8]. Thus, we made a cross-validation per annotated file where also data from the same subject was excluded for training. We performed those two evaluations to show the impact of the training data to our method, which is also a limiting factor for the applicability of machine learning. For the proposed approach, the training uses all remaining data and simulated data with a Power8 server. This was due to large amounts of RAM is necessary to hold all of the data stream manipulations. In addition, we evaluated these data streams in parallel to compute the optimizations in equation 1 and 2.

All of the evaluated state-of-the-art algorithms are configured for offline use. Since all algorithms except IBDT are for offline use only, this means that we did not deactivate the data smoothing, which makes it only applicable offline. For a fair comparison, the trained classifiers also consider subsequent detections and velocities in the data. Therefore, the trained detectors are only applicable offline too.

Algorithm	Limitations
EV	*Cannot detect smooth pursuits.*
IBDT	*Cannot detect Errors and post saccadic movement.*
	Problems with error in the data.
LS	*Cannot detect post saccadic movement.*
I2MC	*Cannot detect Errors and post saccadic movement.*
	Problems if smooth pursuits are present.

Table 1: Overview of the limitations of the state of the art algorithms.

Preliminary stated limitations by the authors of the state of the art algorithms are shown in table 1. As can be seen, not all algorithms can detect all the evaluated types. Therefore, we decided two use two statistics to determine the quality of the algorithms. The first statistic is recall ($TP/(TP + FN)$), which is not influenced by the number of different types that could be incorrectly classified as another type (for example a smooth pursuit as a fixation). Recall specifies how much of one class is correctly classified. The second statistic is precision ($TP/(TP + FP)$). This value is influenced by the number of different types, for instance, a smooth pursuit that is classified as a fixation counts to the false positives (FP). Precision is used to evaluate how reliable the detection per type is.

Data	Alg.	Recall			
		Fixation	*Saccade*	*Pursuit*	*PSM*
DS-AND	EV	0.61	0.73	0	0.02
	IBDT	0.78	0.44	0.61	0
	LS	0.92	0.90	0.16	0
	I2MC	0.02	0.96	0	0
DS-IBDT	EV	0.32	0.35	0	-
	IBDT	0.98	0.56	0.84	-
	LS	0.95	0	0.07	-
	I2MC	0.92	0.1	0	-

Table 2: Recall for each eye movement type <u>without errors</u> in the data. PSM stands for post saccadic movement.

Table 2 shows the recall of all state-of-the-art algorithms on the data where all errors are removed from the data (for IBDT they where marked with setting the position to zero). The rest of the data with unannotated samples was given as input to the algorithms. The IBDT does not perform as good as in the original paper, which is because we did not use the provided data class which smooths out errors by replacing them with previously recorded positions. It was not used to keep the evaluation equal between all algorithms.

Data	**Alg.**	**Recall**				
		Fixation	*Saccade*	*Pursuit*	*Noise*	*PSM*
DS-AND	EV	0.61	0.73	0	0.94	0.02
	IBDT	0.65	0.35	0.63	0	0
	LS	0.91	0.88	0.15	0.13	0
	I2MC	0.02	0.95	0	0	0
	Proposed	0.79	0.85	0.69	0.67	0.78
	Proposed (all)	0.92	0.92	0.89	0.94	0.97
DS-IBDT	EV	0.18	0.25	0	1.0	-
	IBDT	0.97	0.28	0.84	0	-
	LS	0.95	0	0.06	0	-
	I2MC	0.92	0.10	0	0	-
	Proposed	0.94	0.91	0.89	0.65	-
	Proposed (all)	0.97	0.99	0.92	0.98	-

Table 3: Recall for each eye movement type with errors in the data. PSM stands for post saccadic movement.

Table 3 shows the results giving the algorithms the unmodified data. As can be seen for the algorithm EV [23], the error has no impact to the algorithm. This is due to the construction of the algorithm, which tries in the first step to remove the noise by separating it from saccades. Overall, LS [19] seems to be the best state of the art algorithm on the high-frequency data set DS-AND [1](1250Hz) for saccades and fixations. For different recording frequencies as it is in the DS-IBDT [27](30Hz), the algorithm fails to estimate its parameters. The same can be seen for EV, which still detects the noise very well. In table 4, it can be seen that exactly this is the main problem of EV in the DS-IBDT dataset, since it annotates most of the data points as noise. In general, it has to be mentioned that the annotation in DS-IBDT slightly differs from DS-AND. In DS-IBDT, saccades are annotated after the velocity peek. This is because of the low-frequency recording of the data and a disadvantage especially for I2MC [9], which detects the saccades too early. This issue has to be especially highlighted because different annotators assign different labels to data samples. Therefore, algorithms have also to adapt to the researcher-wanted annotation. The proposed approach is based on machine learning, which is capable of learning from samples. Meaning, based on the annotations given in the training data, the detector learns how to label the data. As can be seen in table 3, the proposed approach outperforms the state of the art algorithms. It has to be especially noted that the impact of the amount of training data (proposed vs proposed (all)) is not as high for the low-frequency data as for the data set DS-AND with 1250 Hz. This is due to the smoothing of the data through the low sampling rate itself. The proposed approach was capable of detecting all data types and also the noise in the data which also opens up the possibility of detecting misdetections in images through the movement of the eye.

Data	**Alg.**	**Percision**				
		Fixation	*Saccade*	*Pursuit*	*Noise*	*PSM*
DS-AND	EV	0.68	0.37	0	0.73	0.03
	IBDT	0.72	0.70	0.35	0	0
	LS	0.82	0.33	0.63	0.06	0
	I2MC	0.09	0.08	0	0	0
	Proposed	0.79	0.65	0.82	0.59	0.32
	Proposed (all)	0.94	0.88	0.94	0.94	0.51
DS-IBDT	EV	0.78	0.31	0	0.01	-
	IBDT	0.93	0.73	0.76	0	-
	LS	0.77	0	0.23	0	-
	I2MC	0.76	0.071	0	0	-
	Proposed	0.98	0.86	0.89	0.61	-
	Proposed (all)	0.98	0.92	0.95	0.75	-

Table 4: The precision for each eye movement type with errors in the data. PSM stands for post saccadic movement.

Table 4 shows the precision of each algorithm per eye movement type. As can be seen for the proposed approach the post-saccadic movement in the high-frequency data set is still difficult to detect even with large amounts of data (proposed all). In our experiment, this also comes from the generated data since the simulator is not able to generate post saccadic movement. We, therefore, generated two saccades successively the latter being approximately one-third of the first one. This is not the same as the real post saccadic movement but even in this simple approximation, it helped to improve the detection rate.

5 CONCLUSION

We proposed a novel eye movement detection approach that is based on machine learning. It is capable of training detectors for specific eye movements and extendable to new eye movement types. The detectors are capable of outperforming the state-of-the-art and adaptable to new challenges from new eye trackers. In addition, the detectors can be trained for offline and online analysis, enabling a second validation and refinement stage for eye movement detection. Future research will go into the training of detectors, which are independent of the sampling frequency as far as possible. This could help in reusing detectors even if the eye tracker has changed, which could happen if the used technology is outdated.

ACKNOWLEDGMENTS

This research was supported by an IBM Shared University Research Grant including an IBM PowerAI environment. We especially thank our partners Benedikt Rombach, Martin Mähler and Hildegard Gerhardy from IBM for their expertise and support.

REFERENCES

[1] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. 2017. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods* 49, 2 (2017), 616–637.
[2] Christian Braunagel, David Geisler, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2017. Online Recognition of Driver-Activity Based on Visual Scanpath Classification. *IEEE Intelligent Transportation Systems Magazine* 9, 4 (2017), 23–36.

[3] Frans W Cornelissen, Enno M Peters, and John Palmer. 2002. The Eyelink Toolbox: eye tracking with MATLAB and the Psychophysics Toolbox. *Behavior Research Methods, Instruments, & Computers* 34, 4 (2002), 613–617.
[4] Leandro L Di Stasi, Michael B McCamy, Stephen L Macknik, James A Mankin, Nicole Hooft, Andrés Catena, and Susana Martinez-Conde. 2014. Saccadic eye movement metrics reflect surgical residents' fatigue. *Annals of surgery* 259, 4 (2014), 824–829.
[5] Andrew T Duchowski. 2002. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers* 34, 4 (2002), 455–470.
[6] Ralf Engbert and Reinhold Kliegl. 2003. Microsaccades uncover the orientation of covert attention. *Vision research* 43, 9 (2003), 1035–1045.
[7] Ralf Engbert and Konstantin Mergenthaler. 2006. Microsaccades are triggered by low retinal image slip. *Proceedings of the National Academy of Sciences* 103, 18 (2006), 7192–7197.
[8] Wolfgang Fuhl, Thiago Santini, Thomas Kuebler, Nora Castner, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2018. Eye movement simulation and detector creation to reduce laborious parameter adjustments. *eprint arXiv:1804.00970* (2018).
[9] Roy S Hessels, Diederick C Niehorster, Chantal Kemner, and Ignace TC Hooge. 2017. Noise-robust fixation detection in eye movement data: Identification by two-means clustering (I2MC). *Behavior research methods* 49, 5 (2017), 1802–1823.
[10] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. 2011. *Eye tracking: A comprehensive guide to methods and measures.* OUP Oxford.
[11] Ignace Hooge, Marcus Nyström, Tim Cornelissen, and Kenneth Holmqvist. 2015. The art of braking: Post saccadic oscillations in the eye tracker signal decrease with increasing saccade size. *Vision research* 112 (2015), 55–67.
[12] Qiang Ji, Zhiwei Zhu, and Peilin Lan. 2004. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE transactions on vehicular technology* 53, 4 (2004), 1052–1068.
[13] Enkelejda Kasneci, Gjergji Kasneci, Thomas C Kübler, and Wolfgang Rosenstiel. 2014. The applicability of probabilistic methods to the online recognition of fixations and saccades in dynamic scenes. In *Proceedings of the Symposium on Eye Tracking Research and Applications.* ACM, 323–326.
[14] Reinhold Kliegl and Richard K Olson. 1981. Reduction and calibration of eye monitor data. *Behavior Research Methods* 13, 2 (1981), 107–111.
[15] Oleg V Komogortsev, Denise V Gobert, Sampath Jayarathna, Do Hyong Koh, and Sandeep M Gowda. 2010. Standardization of automated analyses of oculomotor fixation and saccadic behaviors. *IEEE Transactions on Biomedical Engineering* 57, 11 (2010), 2635–2645.
[16] Oleg V Komogortsev and Javed I Khan. 2009. Eye movement prediction by oculomotor plant Kalman filter with brainstem control. *Journal of Control Theory and Applications* 7, 1 (2009), 14–22.
[17] Thomas C Kübler, Colleen Rothe, Ulrich Schiefer, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2017. SubsMatch 2.0: Scanpath comparison and classification based on subsequence frequencies. *Behavior Research Methods* 49, 3 (2017), 1048–1064.
[18] Linnéa Larsson, Marcus Nyström, Richard Andersson, and Martin Stridh. 2015. Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control* 18 (2015), 145–152.
[19] Linnéa Larsson, Marcus Nyström, and Martin Stridh. 2013. Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *IEEE Transactions on Biomedical Engineering* 60, 9 (2013), 2484–2493.
[20] R John Leigh and David S Zee. 2015. *The neurology of eye movements.* Vol. 90. Oxford University Press, USA.
[21] James G May, Robert S Kennedy, Mary C Williams, William P Dunlap, and Julie R Brannan. 1990. Eye movement indices of mental workload. *Acta psychologica* 75, 1 (1990), 75–89.
[22] Marcus Nyström, Richard Andersson, Måns Magnusson, Tony Pansell, and Ignace Hooge. 2015. The influence of crystalline lens accommodation on post-saccadic oscillations in pupil-based eye trackers. *Vision research* 107 (2015), 1–14.
[23] Marcus Nyström and Kenneth Holmqvist. 2010. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods* 42, 1 (2010), 188–204.
[24] C. Rashbass. 1961. The relationship between saccadic and smooth tracking eye movements. *The Journal of Physiology* 159, 2 (1961), 326–338.
[25] Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124, 3 (1998), 372.
[26] Dario D Salvucci and Joseph H Goldberg. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the Symposium on Eye Tracking Research and Applications.* ACM, 71–78.
[27] Thiago Santini, Wolfgang Fuhl, Thomas Kübler, and Enkelejda Kasneci. 2016. Bayesian identification of fixations, saccades, and smooth pursuits. In *Proceedings of the Symposium on Eye Tracking Research and Applications.* ACM, 163–170.
[28] John A Stern, Donna Boyer, and David Schroeder. 1994. Blink rate: a possible measure of fatigue. *Human factors* 36, 2 (1994), 285–297.
[29] Juan Tabernero and Pablo Artal. 2014. Lens oscillations in the human eye. Implications for post-saccadic suppression of vision. *PloS one* 9, 4 (2014), e95764.
[30] Enkelejda Tafaj, Gjergji Kasneci, Wolfgang Rosenstiel, and Martin Bogdan. 2012. Bayesian online clustering of eye movement data. In *Proceedings of the Symposium on Eye Tracking Research and Applications.* ACM, 285–288.
[31] Ralf van der Lans, Michel Wedel, and Rik Pieters. 2011. Defining eye-fixation sequences across individuals and tasks: the Binocular-Individual Threshold (BIT) algorithm. *Behavior Research Methods* 43, 1 (2011), 239–257.
[32] Giacomo Veneri, Pietro Piu, Pamela Federighi, Francesca Rosini, Antonio Federico, and Alessandra Rufa. 2010. Eye fixations identification based on statistical analysis-case study. In *Cognitive Information Processing (CIP), 2010 2nd International Workshop on.* IEEE, 446–451.
[33] Giacomo Veneri, Pietro Piu, Francesca Rosini, Pamela Federighi, Antonio Federico, and Alessandra Rufa. 2011. Automatic eye fixations identification based on analysis of variance and covariance. *Pattern Recognition Letters* 32, 13 (2011), 1588–1593.
[34] Frances C Volkmann, Lorrin A Riggs, and Robert K Moore. 1980. Eyeblinks and visual suppression. *Science* 207, 4433 (1980), 900–902.
[35] Heino Widdel. 1984. Operational problems in analysing eye movements. *Advances in psychology* 22 (1984), 21–29.