IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 19, 2019, accepted October 25, 2019, date of publication November 4, 2019, date of current version November 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951506

Evaluation of Ten Open-Source Eye-Movement Classification Algorithms in Simulated Surgical Scenarios

GONCA GOKCE MENEKSE DALVEREN[1,2] AND NERGIZ ERCIL CAGILTAY[3]
[1]Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjøvik, Norway
[2]Department of Information Systems Engineering, Atilim University, 06830 Ankara, Turkey
[3]Department of Software Engineering, Atilim University, 06830 Ankara, Turkey

Corresponding author: Gonca Gokce Menekse Dalveren (goncagmenekse@gmail.com)

ABSTRACT Despite providing several insights into visual attention and evidence regarding certain brain states and psychological functions, classifying eye movements is a highly demanding process. Currently, there are several algorithms to classify eye movement events which use different approaches. However, to date, only a limited number of studies have assessed these algorithms under specific conditions, such as those required for surgical training programmes. This study presents an investigation of ten open-source eye-movement classification algorithms using the Eye Tribe eye-tracker. The algorithms were tested on the eye-movement records obtained from 23 surgical residents, who performed computer-based surgical simulation tasks under different hand conditions. The aim was to offer data for the improvement of surgical training programmes. According to the results, due to the different classification methods and default threshold values, the ten algorithms produced different results. Considering the fixation duration, the only common event for all of the investigated algorithms, the binocular-individual threshold (BIT) algorithm resulted in a different clustering compared to the other algorithms. Based on the other set of common events, three clusters were determined by eight algorithms (except BIT and event detection (ED)), distinguishing dispersion-based, velocity-based and modified versions of velocity-based algorithms. Accordingly, it was concluded that dispersion-based and velocity-based algorithms provided different results. Additionally, as it individually specifies the threshold values for the eye-movement data, when there is no consensus about the threshold values to be set, the BIT algorithm can be selected. Especially for such cases like simulation-based surgical skill-training, the use of individualised threshold values in the BIT algorithm can be more beneficial in classifying the raw eye data and thus evaluating the individual progress levels of trainees based on their eye movement behaviours. In conclusion, the threshold values had a critical effect on the algorithm results. Since default values may not always be suitable for the unique features of different data sets, guidelines should be developed to indicate how the threshold values are set for each algorithm.

INDEX TERMS Eye-movement classification algorithms, eye-movement events, eye-tracking.

I. INTRODUCTION

Today's eye-movement tracking technology offers several benefits, and thus is widely used in various fields, including neurology, psychology, ophthalmology, and commercial areas. Classifying eye movements is crucial for understanding visual attention and providing evidence regarding certain brain states and psychological functions [1]–[4]. Such evidence can later be used for diagnoses, treatment and training purposes [5]–[9]. Eye-movement classification is also important for clinical applications related to important health conditions and diseases, such as Alzheimer's disease [10], [11], HIV-1 infections presenting with eye-movement dysfunction [12], and schizophrenia [13], [14]. Commercial purposes include Web navigation, shopping, and human-computer interaction [3], [4], [15], [16]. Besides, several studies have been undertaken to investigate eye movements in order to support surgical training programmes, specifically to improve surgical simulations and needle insertion simulations to represent real-life environments, such as

The associate editor coordinating the review of this manuscript and approving it for publication was Sunith Bandaru.

soft tissues [17]–[21]. These improvements on the techniques used for surgical simulations have several potential benefits for surgical training programmes in terms of providing support for skill assessment procedures [22] and helping candidate surgeons acquire the necessary skills [23]. Additionally, there is some evidence showing the benefits of better understanding the eye-movement events of surgical residents while they are performing surgical tasks in computer-based simulation environments [24]. The classification of basic eye-movement events from noisy eye data is essential, particularly for researchers using eye trackers to record data in their studies [2], [25]–[27]. Research using static stimuli mostly focuses on fixations that refer to slow periods when the eye is nearly still, and saccades that occur when the eye makes rapid shifting movements between different positions. The third commonly examined eye movement event is smooth pursuit, which is tracking of dynamic stimuli, such as moving objects in a visual scene [28]. This event produces velocities that are inconsistently handled by algorithms [29]. Therefore, unlike fixations and saccades, there is no clear classification of smooth pursuit eye movements, and their analysis depends on the threshold values of the algorithms used [29].

It is up to researchers whether to manually identify eye-movement events or use a commercially or freely available algorithm for such classification. However, the use of classification algorithms is now accepted as the only practical solution for this purpose considering the time-consuming nature of manual assessments [29].

In the literature, there are not many studies that comparatively evaluated eye-movement classification algorithms. Therefore, in this study, different algorithms were tested on the eye-movement data collected from 23 surgeons while they were performing four virtual simulation scenarios in a 3D dynamic and interactive environment under different hand conditions, namely dominant hand, non-dominant hand, and both hands.

A. CURRENT EYE-MOVEMENT CLASSIFICATION ALGORITHMS

Several eye-movement classification algorithms are currently in use; however, some are not open-source or are only commercially available as part of software suites developed by companies manufacturing eye-tracking devices, making it difficult to evaluate them. There are also some machine learning algorithms used to classify data [30] into eye movement events. However, in this study, for convenience, ten open-source eye-movement classification algorithms implemented with MATLAB were selected based on the criteria of being up-to-date and independent of any eye-tracker device (See Appendix-A) [29].

Eye-movement algorithms classify eye data by considering different features gathered by eye tracker devices. There are different types of algorithms, mainly categorised as velocity-based and dispersion-based, to classify eye-movement events, such as fixations, saccades, and smooth pursuits. Velocity-based algorithms include Velocity Threshold Identification (I-VT), Hidden Markov Model Identification (I-HMM), and Kalman Filter Identification (I-KF), and examples of dispersion-based algorithms are Dispersion Threshold Identification (I-DT) and Minimum Spanning Tree Identification (I-MST) [31]. Furthermore, Velocity and Velocity Threshold Identification (I-VVT) is the modified version of I-VT to distinguish smooth pursuits from fixations, and similarly, Velocity and Movement Pattern Identification (I-VMT) utilizes the velocity threshold to classify saccades and investigates the eye-movement samples to separate smooth pursuits from fixations. Lastly, Velocity and Dispersion Threshold Identification (I-VDT), a modified combination of velocity-based and dispersion-based algorithms, is used for ternary classifications [31].

I-VT is based on the velocity of eye movement to separate fixation events from saccade events and employs a velocity threshold to classify fixations and saccades. If the velocity of a sample is below the threshold, then the algorithm defines it as a fixation, and if it above the threshold, then the event is classified as a saccade. This velocity threshold principle is also the basis for other algorithms [31]; for example, I-VVT discriminates between different eye-movement events through a filter function that eliminates noisy saccade-like events based on minimum amplitude and duration values [31]. I-VMT evaluates the movement patterns to distinguish smooth pursuits from fixations. Such movement patterns are examined in a temporal window, where the magnitude of movement is computed by analysing the angles created by each pair of adjacent positional points and the horizontal coordinate axis. Once the value representing the magnitude of motion is calculated, the values above the threshold are marked as smooth pursuits and those below the threshold as fixations [31].

I-VDT is also used for the classification of fixations, saccades, and smooth pursuits. Similar to the I-VVT and I-VMT algorithms, I-VDT also separates the saccades first and then distinguishes smooth pursuits from fixations using an adapted I-DT method [31]. Similar to the I-VT algorithm, I-HMM is based on velocity but it has two additional algorithms: the first re-classifies fixations and saccades according to probabilistic parameters, and the second updates the parameters [1]. I-KF is a recursive predictor that computes future states by estimating a series of dynamic system states from noisy measurements [32]. Since the actual data is usually noisy and may cause data loss, I-KF minimizes the difference between the state of the estimated system and that of the real system. It only requires the previous time step and estimated new conditions to calculate the estimate of the new event [32]. If the value is less than the set threshold and the minimum time threshold is met, the event is classified as a fixation, and if the value is above the threshold, it is considered a saccade [32].

I-DT, one of the dispersion-based algorithms, is commonly used to classify eye-movements into fixations and saccades using the x and y coordinates of the eye and two thresholds, namely maximum and minimum fixed-time thresholds [1].

I-MST, another dispersion-based algorithm takes a predetermined number of eye position points to create a MST [33], which is defined as a spanning tree with the least distance among all spanning trees in a node set. I-MST breaks MST into segments and thresholds based on predetermined distances [33]. The advantage of using an I-MST is that this algorithm can correctly identify the anchor points if a large part of the signal is missing. As a result of this property, I-MST is claimed to be a highly flexible and controllable eye-movement detection tool [33].

Adaptive event detection (ED) is a velocity-based algorithm that alters the velocity threshold based on the noise level of the subject. It also distinguishes post-saccadic releases, also known as 'glissades', from fixations and saccades [34]. As a different eye-movement, a glissade is a wobbling movement at the end of saccades. The ED algorithm was developed to classify fixation, saccade and glissade eye-movement events and provide graphical representations of these events [34]. Another velocity-based algorithm is Binocular-Individual Threshold (BIT) [35], which identifies fixations in the movement data of both eyes based on individual-specific thresholds. BIT has advantages over other similar algorithms in that it contains binocular viewing and uses the information about fixations and co-variations between the movements of both eyes to identify saccades; it estimates rather than pre-sets the velocity threshold to identify fixations and saccades, thus allowing the threshold to vary between eye-movement directions, tasks, and individuals. In addition, each record exceeding the threshold value contains stochasticity, which is spontaneous in eye-movements, and therefore prevents inaccurately labelling events as saccades [35]. Another important feature of BIT is that it is independent of machine and sampling frequency; thus, it can be easily adapted to the data from varying eye-trackers with different sensitivity and sampling frequencies [35].

B. LIMITATIONS OF CURRENT EYE-MOVEMENT CLASSIFICATION ALGORITHMS

The need for a single algorithm to be used in all systems and the presence of many algorithms addressing the same problem imply that eye-movement classification is not a mundane issue, and assessing the performance of different algorithms is an important undertaking [29]. Notably, selecting the most appropriate algorithm means that a thorough evaluation method has to be designed [2]. Since this study is not the first that attempted to evaluate algorithm performance, it is necessary to consider the benefits and drawbacks of the previously established methods.

One of the approaches has been to establish an optimal or rational relationship between stimuli and an individual's viewing behaviour. For instance, researchers offered a trial to the participants in order to look at a single moving target that jumped a number of times [1]. Given the known number of jumps, positions, and amplitudes, it is possible to calculate how an ideal eye-movement behaviour should appear; then, the gaze data parsed by the algorithms is compared to the ideal viewing pattern. The use of a higher number of algorithm provides more reliable results [1].

The detection of eye-movement events is not a completely resolved problem because there is no consensus on how to evaluate the algorithms. This hinders the further refinement of algorithms as it is not clear whether the differences are due to the algorithms being used or the evaluation process itself [25], [29]. Furthermore, it is not clearly known what we refer to as an event; for example, there is no theoretically motivated threshold for the eye to be classified as a saccade to sufficiently move in a particular direction or to classify anything under it as another event. Classification algorithms focus on a rigorous oculomotor definition of fixations and saccades. Even in the definition of a fully oculomotor eye movement, it is difficult to identify the end point of the fixation and the start point of a smooth pursuit. This point is arbitrary and more or less determined by the sensitivity of the system. Many algorithms require some form of adjustment to be made by the researcher, such as minimum fixation time and saccade speed threshold. If there is an explicit and theoretically applied threshold, then it will already be coded as a constant value in the algorithm. However, setting the thresholds depends on the researcher. It is common knowledge that different parameter values for these algorithms produce different classification results [29], but there is a lack of data on the comparative analysis of many algorithms. Often, a modest assessment is performed while presenting a new algorithm, but in this process, only a limited number of algorithms are examined, and the main focus is on the new algorithm.

C. THRESHOLD VALUES OF EYE-MOVEMENT CLASSIFICATION ALGORITHMS

The algorithms evaluated in the present study were used with only slight changes to their default settings. However, an ideal algorithm does not require the user to set any parameter and automatically adjusts the thresholds for the comprehensive categorisation of all samples of the data stream [29]. Certain parameter values are the same for all algorithms, namely screen size value (1920 $\times$ 1080), distance from the screen (70 cm), and sampling frequency (60 Hz). In the following section, all the remaining parameters and threshold values for each algorithm are listed, and this information is also presented as a table in Appendix-B. Algorithms use saccade detection thresholds for classifying saccades and fixations. The velocity value is computed for each eye position sample, which is then compared to the threshold value. If the sampled velocity is less than the threshold, the corresponding eye-position sample is marked as a fixation; otherwise, it is marked as a saccade. In other words, if the movement speed from an eye position to the next is greater than the threshold value, this means that the eye movement belongs to a saccade event; otherwise, it indicates a fixation or smooth pursuit. After this evaluation, another threshold is used to discriminate between fixation and smooth pursuit events. The fixation detection threshold sets the value of the speed threshold used to distinguish between fixations and smooth pursuits. The speed of the movement from one eye position to the next being larger than this value means that the eye movement is a smooth pursuit; otherwise, it is a fixation.

D. VELOCITY THRESHOLD IDENTIFICATION (I-VT)

The I-VT algorithm uses the saccade detection threshold to classify saccades and fixations. This model, the velocity value is computed for each eye position sample, and then compared to the threshold. If the sampled velocity is less than the threshold, the corresponding eye-position sample is marked as a fixation; otherwise, it is marked as a saccade. The default threshold value is 70°/s. If the movement speed from one eye position to the next is below this value, the eye movement belongs to a fixation event, and if above, it is classified as a saccade.

E. VELOCITY AND VELOCITY THRESHOLD IDENTIFICATION (I-VVT)

The I-VVT algorithm uses two threshold parameters; one for saccade detection and the other for fixation detection. The former is set to 70°/s by default and adjusts the value of the speed threshold used to distinguish between saccades and fixations. If the movement speed from an eye position to the next is greater than this value, the eye movement belongs to a saccade; otherwise, it is a fixation or smooth pursuit. The second threshold (fixation detection), set to 20°/s by default, refers to the speed value to be used to distinguish between fixations and smooth pursuits. If the speed of the movement from one eye position to the next is larger than this value, the eye movement is a smooth pursuit; if smaller, it is a fixation.

F. VELOCITY AND MOVEMENT PATTERN IDENTIFICATION (I-VMT)

The I-VMT algorithm uses three threshold parameters: saccade detection threshold, temporary window length, and range threshold. The saccade detection threshold field on the I-VT tab adjusts the value of the speed value used to differentiate between saccades and fixations. This parameter is set to 70°/s by default. If the movement speed from an eye position to the next is greater than this value, it belongs to a saccade event; otherwise, it is either a fixation or a smooth pursuit, which should be clarified in the next stage of classification. The temporary window length field specifies how much time is needed for dispersion calculation during data processing. The default value of this parameter is 0.5. Lastly, the range threshold value sets the threshold for the distribution of the selected samples in a range of 0 to 1. This parameter is set to 0.1 by default.

G. DISPERSION THRESHOLD IDENTIFICATION (I-DT)

The I-DT algorithm uses two threshold parameters: dispersion duration and dispersion degree. The dispersion duration threshold specifies how much time is required to calculate the distribution during data processing. This field sets the threshold value for the distribution of the selected samples in degrees. If the distribution is less than the specified value, it means that a fixation has been detected; otherwise, the eye movement is a saccade. The default values for this classification are 100 ms for the dispersion duration and 1.35° for the dispersion degree.

H. VELOCITY AND DISPERSION THRESHOLD IDENTIFICATION (I-VDT)

The I-VDT algorithm employs three threshold parameters as saccade detection, dispersion duration, and dispersion. The saccade detection threshold on the I-VT tab adjusts the value of the speed threshold to distinguish between saccades and fixations. If the movement speed from an eye position to the next is greater than this value, the eye movement is a saccade; otherwise, it is either a fixation or a smooth pursuit. In the latter case, further classification is needed. The saccade detection threshold is set to 70°/s by default. The dispersion duration threshold on the I-DT tab specifies how much time is needed to calculate distribution during data processing. The default value of this parameter is 100 ms. The dispersion threshold field on the I-DT tab sets the threshold value for the distribution of the selected samples in degrees. If the distribution is less than this value, it means that a fixation has been detected; otherwise, the eye movement is a saccade. This parameter is set to 1.35° by default.

I. HIDDEN MARKOV MODEL IDENTIFICATION (I-HMM)

The I-HMM algorithm uses three threshold parameters, namely saccade detection threshold, Viterbi sample size, and Baum-Welch reiteration. The saccade detection threshold is identical to the I-VT classifier. The Viterbi sample size specifies the number of samples used by the classifier as a data set. If the threshold value is set too high, there is not enough machine precision to calculate the statistical parameters. The Baum-Welch iteration specifies the number of iterations to be used in the Baum-Welch algorithm. The default values are 70°/s for the saccade detection threshold, 200 for the Viterbi sample size, and 5 for the Baum-Welch iteration.

J. KALMAN FILTER IDENTIFICATION (I-KF)

The I-KF algorithm uses the three parameters of Chi threshold, sampling window size, and deviation. The value for the $\chi 2$-distribution threshold is set by the Chi threshold field. If the $\chi 2$-distribution values are lower than this threshold, then a fixation has been detected; if not, it is a saccade. The sampling window size specifies the number of samples for which the $\chi 2$-distribution is calculated. The deviation field sets the deviation value between the anticipated and computed values for the $\chi 2$-distribution calculation. The default parameter values are 15 for the chi-square threshold, 5 samples for the window size, and 1000 for deviation.

K. MINIMUM SPANNING TREE IDENTIFICATION (I-MST)

The I-MST algorithm contains two threshold parameters: saccade detection threshold and window size. The saccade detection threshold adjusts the distance between two adjacent eye focus positions in degrees. If this distance is less than the threshold, the eye movement is a fixation, and if higher than the threshold value, it indicates a saccade. The window size field sets the number of instances used by the classifier during data processing. If it is set too high, the sensing time for the saccades can be significantly increased. Therefore, it is reasonable to set this value slightly lower than the average fixation duration. The default values for the saccade detection threshold and window size parameters are 0.6° and 200 samples, respectively.

L. ADAPTIVE EVENT DETECTION ALGORITHM (ED)

In this study, the default ED values were used, except the minimum saccade duration that was set at 20 ms because the algorithm was designed for 1250 Hz of data with high precision, and the window length (F) did not scale satisfactorily when using 60 Hz data. Therefore, it was considered to be more appropriate to increase F to the nearest odd integer (F = 3). To do this, there was also a need to increase the value of the minimum saccade duration from 10 to 20 ms.

M. BINOCULAR-INDIVIDUAL THRESHOLD (BIT)

BIT is a parameter-free fixation identification algorithm that automatically identifies relative and individual specific speed thresholds by optimally using the statistical properties of the eye data. To verify the fixations, BIT algorithm uses velocity thresholds. The advantages of this algorithm is the automatic identification of the task and individual specific velocity threshold values and independence from the eye-tracker and sampling frequency. This algorithm can be used with the data recorded by various eye tracker devices.

II. MATERIALS AND METHODS

Prior to the study, the ethical approval from the Human Research Ethics Board of Atilim University was received. All participants were individually informed about the purpose of the study, procedures, and length of experiment. Only volunteers were recruited for the study, and their verbal and written consent was obtained. First, the common eye-movement events of the above-mentioned 10 algorithms were analysed; then the results of the classifications were examined to better understand the commonalities and differences between these algorithms. The following research questions were investigated:

RQ1. Are there any differences between the ten algorithms tested?

RQ2. Do the algorithms form different clusters?

A. SCENARIOS

Four different scenarios simulating different surgical procedures in a virtual environment were designed in accordance with the feedback of expert surgeons from Hacettepe University Medical School with more than 30 years' experience in skull base surgery. Their educational evaluations including these scenarios also show some encouraging results for

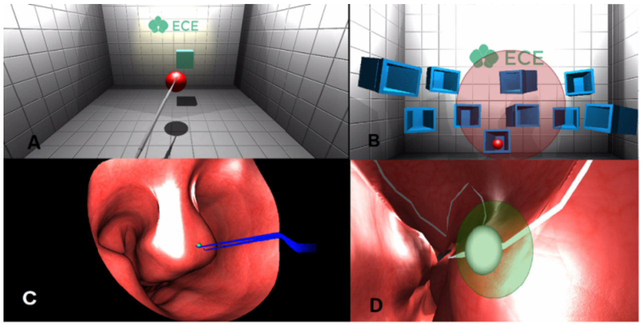

FIGURE 1. Scenarios.

supporting surgical education programmes [23]. Each scenario contained different tasks to be performed to develop different skills among surgical residents, such as eye-hand coordination, fast tracking of objects, depth perception, effective use of surgical instruments, and simultaneous use of both hands. In this research, different hand conditions were also tested. The participants were first asked to perform the tasks using their dominant hand, then with their non-dominant hand, and finally using both hands. In the first scenario (Figure 1: A), the participants were expected to catch a red ball that appeared at random locations in a room displayed on the screen using a surgical instrument. Once the participants caught the red ball, the colour of the ball turned green, and the participants had to move the ball to the cube. The cube also appeared at random locations within the room. This process was repeated ten times.

In the second scenario (Figure 1: B), the participants were asked to use a simulated instrument to touch the red balls that randomly appeared in one of the blue boxes on the screen. Once the participants focused on the red ball at the right angle, the ball exploded. This process was also repeated ten times. In the third scenario (Figure 1: C), the participants were expected to find and remove tumour-like objects from a virtual simulation of the human nose model using an endoscopic instrument. There were ten tumours located in different parts of the nose model. In the last scenario (Figure 1: D), the participants were instructed to continuously follow a moving object (a white ball) on a fixed path inside the simulated nose environment. In order to be able to move the white ball on this path, it was necessary to focus on the ball at a correct angle and distance by controlling the endoscopic instrument through a haptic device.

B. DATA COLLECTION PROCESS

Twenty-eight surgical residents (23 male and 5 female) participated in this study. However, due to the recording failure of the eye tracker, the data of five participants was not recorded, and therefore the analysis was conducted on the data of 23 participants. In Turkey, after completing 12-year primary and secondary education, students are accepted to the medical faculties of universities based on their scores in the central university entrance examination containing multiple-choice questions [36]. The undergraduate medical education lasts six years, after which medical graduates are placed, based on the grades from another centralized exam, in specialisation programmes pre-determined by the Turkish Ministry of Health [36]. The participants of the present study were surgical residents enrolled in the specialisation programmes of neurosurgery and ear-nose-throat (ENT) departments.

The eye-movement data of the surgical residents was recorded using Eye Tribe at 60 Hz sampling frequency. This eye-tracker was chosen as an affordable and potentially valuable tool for this research [37]. Previous studies reported that Eye Tribe significantly differentiated the pupil dilations caused by mental workload [38]. In addition, this eye tracker was presented as a promising tool for researchers investigating human factors [38]. To ensure high accuracy in the eye-movement recordings, the participants were positioned at a fixed distance (70 cm) from the monitor. At the start of each experiment, a calibration procedure containing nine targets was performed for each participant to ensure accuracy. The data was collected while the participants were performing the simulated scenarios. Eye Tribe recorded the eye movement data in the x and y coordinates for the analysis.

A total of four scenarios, one for each surgical operation, were developed using the Unity platform and C# programming language. In surgical operations, physicians have to operate with both hands and use different tools at the same time, meaning that they have to improve their ability to use their dominant and non-dominant hands simultaneously. The novelty of this study is that the four scenarios were performed under different hand conditions (dominant hand, non-dominant hand, and both hands), during which the eye-tracker recorded all the eye movements. To provide more objectivity, half of the participants started to perform the tasks using their dominant hand, and the remaining participants started with their non-dominant hand. The participants used a Geomagic Touch haptic device to perform the tasks in the scenarios under different hand conditions, which provided touch-and-feel feedback concerning the interactions in the simulated environment.

Eye Tribe recorded the eye movement data for each scenario and hand condition for 23 participants in the JSON format [39] as a text document. This data was then converted to the Excel format to be used in the eye-movement classification algorithms. Since each algorithm had different input parameters, the data was adjusted to match the specific requirements of each algorithm. Hence, for each participant, a new Excel file was created to be used in the BIT and ED algorithms, including the values for timestamp and the raw x and y coordinates of the right eye. Since eyes mostly move in the same direction; i.e. when the left eye moves in a certain direction, the right eye would also move in the same direction, only the data from the right eye was used in this study. Considering that the I-DT, I-HMM, I-KF, I-MST, I-VDT, I-VMT, I-VT and I-VVT algorithms expected input parameters in mm, for conversion to mm, the coordinate values saved in pixels were multiplied by a coefficient value of 0.311

TABLE 1. Algorithms and classified events.

Name	Year	VB	DB	FN	FD	SN	SD	SAD	GD	PN	PD	PVD
ED	2010	✓			✓		✓		✓			
I-DT	2000		✓	✓	✓	✓	✓	✓				
I-HMM	2000	✓		✓	✓	✓	✓	✓				
I-KF	2009	✓		✓	✓	✓	✓	✓				
I-MST	2000		✓	✓	✓	✓	✓	✓				
I-VDT	2011	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
I-VMT	2011	✓		✓	✓	✓	✓	✓		✓	✓	✓
I-VT	2000	✓		✓	✓	✓	✓	✓				
I-VVT	2011	✓		✓	✓	✓	✓	✓		✓	✓	✓
BIT	2011	✓		✓	✓	✓						

VB: Velocity based;
DB: Dispersion based;
FN: Fixation Number;
FD: Fixation Duration;
SN: Saccade Number;
SD: Saccade Duration;
SAD: Saccade Amplitude Degree;
GD: Glissade Duration;
PN: Pursuit Number;
PD: Pursuit Duration;
PVD: Pursuit Velocity Degree

(27 inch monitor with 1920 x1080). Then, these eight algorithms were run with the raw x and y coordinates of the right eye in mm obtained for each participant. The results obtained from all algorithms were analysed.

III. RESULTS

To evaluate and compare the different classification methods, each was considered with respect to several characteristics, such as the classified eye-movement events, the methods used for the classification, and the results.

A. ALGORITHMS AND EYE-MOVEMENT EVENTS

As shown in Table 1, the ten algorithms investigated have some common and specific eye-movement events. For instance, glissade duration (GD) is only detected by the ED algorithm whereas the BIT algorithm classifies fixation number (FN), fixation duration (FD) and saccade number (SN) events.

The algorithms I-DT, I-HMM, I-KF, I-MST, I-VDT, I-VMT, I-VT and I-VVT classify FN, FD, SN saccade duration (SD) and saccade amplitude degree (SAD) events. Furthermore, the I-VDT, I-VMT and I-VVT algorithms classify pursuit number (PN), pursuit duration (PD), and pursuit velocity degree (PVD). FD and SD are the eye-movement events that the ED algorithm has in common with the other algorithms. The release dates of the algorithms are presented in Table 1. According to this information, I-VT, I-HMM, I-DT and I-MST were the initial algorithms proposed by [41], and later, I-KF, I-VVT, I-VMT, I-VDT [40], ED [42] and BIT [43] were developed. The investigated algorithms also have different classification methods; for example, I-VT, I-KF, I-HMM, ED and BIT are velocity-based while I-DT and I-MST algorithms are dispersion-based. The I-VVT and I-VMT algorithms, which are modified versions of the I-VT velocity-based algorithm, can identify smooth pursuit eye-movement events. Another algorithm that is able to detect smooth pursuit events is the I-VDT algorithm, which is a modified version of the velocity-based and dispersion-based algorithms.

TABLE 2. Friedman test results for the first scenario under the dominant-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	4.1	6.2	8.0	8.5	1.3			
I-HMM	4.5	7.1	4.0	6.1	6.1			
I-KF	1.8	5.9	4.6	8.3	5.9			
I-MST	1.8	8.9	2.3	2.3	2.3			
I-VDT	7.2	4.2	3.5	3.8	5.4	1.3	1.0	1.8
I-VMT	3.8	2.3	3.1	3.3	6.7	1.7	2.9	1.2
I-VT	5.2	6.7	5.2	4.9	4.9			
I-VVT	8.4	2.7	7.2	6.8	3.5	3.0	2.1	3.0
BIT	8.2	10.0	7.1					
	$x^2$154.56 p<0.001	192.89	102.42	168.25	103.35	37.08	42.35	39.91

TABLE 3. Friedman test results for the first scenario under the non-dominant hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	3.4	6.5	7.4	7.9	2.0			
I-HMM	5.1	6.7	5.9	6.8	5.2			
I-KF	2.0	5.5	4.4	8.5	7.0			
I-MST	1.7	8.9	2.1	2.5	2.5			
I-VDT	6.9	4.1	3.5	4.0	5.4	1.2	1.0	1.8
I-VMT	4.3	2.2	2.7	3.2	5.7	1.8	3.0	1.2
I-VT	5.1	7.2	4.7	4.8	4.5			
I-VVT	7.9	2.9	6.6	6.5	3.6	3.0	2.0	3.0
BIT	8.7	10.0	7.9					
	$x^2$151.43 p<0.001	194.32	112.50	163.00	79.59	39.39	46.00	39.39

B. DIFFERENCES BETWEEN THE CLASSIFICATION RESULTS OF THE ALGORITHMS

All data analysis was carried out using SPSS for Windows software package (version 23; IBM Corporation, New York, USA) at the 95% confidence level. Since the normality assumptions were not satisfied and the sample size was 23, a non-parametric analysis technique, the Friedman test, was used [40], which is an alternative to bi-directional variance analysis for determining the differences between two major masses [40]. This technique was used to determine the differences between the classification results of the ten algorithms for each eye-movement event. According to the results, there were significant differences for all eye-movement events ($p < 0.001$). The mean ranks of the Friedman test for each algorithm and eye-movement event are given for each scenario in Tables 2 to 13.

C. RESULTS OF HIERARCHICAL CLUSTERING

FD was the only common eye-movement event among all algorithms (Table 1). A hierarchical clustering method was applied to provide a better understanding of clustering among the classification results of the ten algorithms. According to the results of hierarchical clustering based on the mean FD for all algorithms and hand conditions, two clusters were identified (Figure 2): Cluster-1 consisting of the BIT algorithm and Cluster-2 containing the remaining nine algorithms. The measured mean values of FD for Cluster-1 and Cluster-2 are given in Table 14.

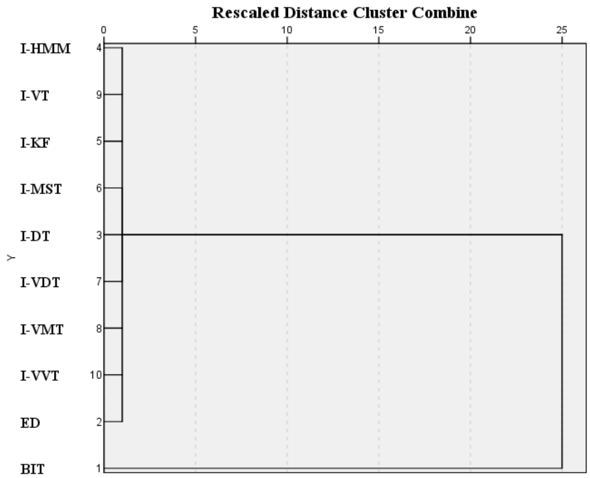

FIGURE 2. Dendrogram of all algorithms using ward's linkage method.

TABLE 4. Friedman test results for the first scenario under the both-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	5.8	5.3	7.8	8.7	1.6			
I-HMM	4.8	7.6	7.8	5.4	4.8			
I-KF	2.8	6.1	2.6	7.8	6.9			
I-MST	1.8	9.0	1.5	2.2	5.1			
I-VDT	6.3	4.1	3.9	4.0	5.3	1.8	1.3	2.0
I-VMT	2.3	2.3	3.1	3.3	6.3	1.3	3.0	1.1
I-VT	5.7	7.0	5.3	5.8	3.5			
I-VVT	6.4	2.9	6.1	6.8	2.5	2.9	1.7	2.9
BIT	9.0	9.6	6.8					
	$x^2$134.06 p<0.001	189.89	135.88	163.87	91.98	32.09	37.13	38.35

TABLE 5. Friedman test results for the second scenario under the dominant-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	5,7	5.1	8.3	8.8	1.4			
I-HMM	6.8	7.7	3.2	6.7	7.0			
I-KF	5.1	6.1	4.0	8.2	6.5			
I-MST	2.8	9.0	1.1	2.0	2.7			
I-VDT	3.8	3.8	4.3	3.9	5.4	1.3	2.0	1.9
I-VMT	1.3	2.5	2.7	3.1	6.6	1.8	3.0	1.1
I-VT	7.8	7.2	6.1	5.0	3.7			
I-VVT	2.5	2.7	7.3	6.3	2.8	2.9	1.0	3.0
BIT	9.0	10.0	8.0					
	$x^2$165.74 p<0.001	202.23	1860.72	181.00	123.05	28.26	46.00	40.78

Since the number of common eye-movement events was high for all algorithms except BIT and ED (Table 1), these two algorithms were excluded and the cluster analysis was repeated with the five common eye-movement events (FN, FD, SN, SD and SAD) for the remaining eight algorithms (I-DT, I-HMM, I-KF, I-MST, I-VDT, I-VMT, I-VT and I-VVT). The results of this analysis revealed three clusters: Cluster-1 containing I-DT and I-VDT, Cluster-2 consisting of I-HMM, I-KF, I-MST and I-VT, and Cluster-3 comprising I-VMT and I-VVT. The dendrogram of these clusters is presented in Figure 3.

TABLE 6. Friedman test results for the second scenario under the non-dominant hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	5.1	4.9	8.7	8.9	1.4			
I-HMM	6.9	7.8	2.7	6.6	7.3			
I-KF	5.7	6.1	3.9	8.0	6.6			
I-MST	3.8	9.0	1.0	2.0	3.1			
I-VDT	3.1	3.7	4.6	4.0	4.8	1.8	2.0	2.0
I-VMT	1.4	2.9	3.0	3.0	6.2	1.3	3.0	1.0
I-VT	7.8	7.2	6.3	5.1	3.8			
I-VVT	2.2	2.5	7.4	6.3	2.7	2.9	1.0	3.0
BIT	9.0	10.0	7.3					
x^2	168.60	201.79	162.38	180.89	119.29	30.02	46.00	42.09
	p<0.001							

TABLE 7. Friedman test results for the second scenario under the both-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	6.7	5.0	8.4	9.0	1.1			
I-HMM	4.8	7.4	4.2	6.7	5.3			
I-KF	3.1	6.4	4.2	7.9	6.0			
I-MST	1.1	9.0	1.1	2.0	4.5			
I-VDT	5.9	4.0	3.2	3.7	6.3	1.9	1.2	2.0
I-VMT	2.1	2.2	2.5	3.3	6.9	1.1	3.0	1.0
I-VT	5.2	7.2	5.8	5.1	3.6			
I-VVT	7.1	2.8	7.1	6.3	2.3	3.0	1.8	3.0
BIT	9.0	10.0	8.5					
x^2	1160.62	202.02	168.89	179.52	110.45	42.00	38.17	46.00
	p<0.001							

TABLE 8. Friedman test results for the third scenario under the dominant-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	7.2	5.0	8.2	9.0	1.2			
I-HMM	5.7	7.2	3.6	6.8	6.9			
I-KF	4.7	6.0	2.3	8.0	7.8			
I-MST	2.9	9.0	1.8	2.3	2.6			
I-VDT	5.1	4.0	4.4	3.8	5.0	1.9	1.8	2.0
I-VMT	1.0	2.3	3.0	2.9	6.1	1.1	3.0	1.0
I-VT	6.1	7.9	6.0	5.0	3.7			
I-VVT	3.6	2.7	7.0	6.2	2.7	3.0	1.2	3.0
BIT	8.7	9.8	8.7					
x^2	132.25	203.67	160.82	179.70	145.82	43.29	40.51	44.09
	p<0.001							

TABLE 9. Friedman test results for the third scenario under the non-dominant hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	6.2	5.2	7.9	8.7	1.2			
I-HMM	6.1	7.2	3.5	6.2	6.6			
I-KF	3.8	5.8	2.7	8.2	7.3			
I-MST	2.6	9.0	1.3	2.1	2.7			
I-VDT	4.4	4.1	4.2	3.8	5.5	1.9	1.8	2.0
I-VMT	1.2	2.2	3.5	3.2	6.0	1.2	3.0	1.0
I-VT	6.8	7.7	6.0	5.1	3.6			
I-VVT	4.9	2.8	7.0	6.8	3.0	3.0	1.2	2.9
BIT	9.0	10.0	8.8					
x^2	140.04	202.29	161.36	178.63	126.57	38.26	39.39	42.09
	p<0.001							

IV. DISCUSSION

The results of this study show that as each algorithm uses different methods of classification, the threshold values are

TABLE 10. Friedman test results for the third scenario under the both-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.0		1.0				
I-DT	6.6	5.0	8.0	8.9	1.2			
I-HMM	5.8	7.7	3.1	6.2	6.5			
I-KF	4.0	6.0	2.7	8.1	7.1			
I-MST	2.8	9.0	1.3	2.1	2.8			
I-VDT	5.0	4.0	4.4	3.9	5.5	1.8	1.8	2.0
I-VMT	1.0	2.1	3.5	3.0	6.1	1.2	3.0	1.0
I-VT	7.0	7.3	6.1	5.0	3.9			
I-VVT	3.7	3.0	7.2	6.8	2.9	3.0	1.2	3.0
BIT	9.0	10.0	8.7					
x^2	143.67	205.54	166.14	181.44	119.00	35.63	38.17	44.09
	p<0.001							

TABLE 11. Friedman test results for the fourth scenario under the dominant-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.00		1.00				
I-DT	7.28	5.13	8.17	8.83	1.22			
I-HMM	4.63	6.91	4.09	6.85	5.43			
I-KF	2.89	6.17	3.57	7.80	6.13			
I-MST	1.39	9.00	1.41	2.17	5.22			
I-VDT	5.83	3.96	4.43	3.91	5.15	1.91	1.87	1.98
I-VMT	2.20	2.22	2.46	2.96	6.50	1.15	3.00	1.02
I-VT	5.48	7.70	5.59	5.00	3.83			
I-VVT	6.30	2.91	7.17	6.48	2.52	2.93	1.13	3.00
BIT	9.00	10.00	8.11					
x^2	152.55	200.96	148.23	176.30	90.67	37.62	40.78	45.52
	p<0.001							

TABLE 12. Friedman test results for the fourth scenario under the non-dominant hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.00		1.00				
I-DT	7.28	4.96	8.87	8.96	1.26			
I-HMM	4.46	7.33	4.87	6.87	5.67			
I-KF	3.09	6.57	4.00	7.72	6.22			
I-MST	1.35	8.98	1.39	2.30	2.85			
I-VDT	6.30	3.78	3.46	3.93	5.65	2.00	1.91	2.00
I-VMT	2.33	2.87	2.41	3.02	7.00	1.00	2.91	1.00
I-VT	5.39	7.13	5.59	4.93	4.33			
I-VVT	5.85	2.39	7.28	6.26	3.02	3.00	1.17	3.00
BIT	8.96	10.00	7.13					
x^2	149.38	2199.71	150.76	173.97	105.63	46.00	35.04	46.00
	p<0.001							

important and need to be set carefully. In addition, eye trackers, computer properties, and algorithm thresholds used in classification vary. Open-source algorithms should be adaptable to work with data obtained from different types of hardware. Yet, it is challenging, particularly for researchers without software development background, to obtain, extract and parse data from different devices to use with open-source eye-movement classification algorithms.

In order to make the ED algorithm workable, it is necessary to have experience in software development to understand the code. Since the algorithm is designed for a 1250 Hz eye-tracker, it needs to be modified for the data received from eye-trackers operating at different sampling rates/frequencies. Accordingly, in this study, the code was adapted to the eye data acquired from Eye Tribe with a 60 Hz sampling frequency.

TABLE 13. Friedman test results for the fourth scenario under the both-hand condition.

ALGORITHM	MEAN RANKS							
	FN	FD	SN	SD	SAD	PN	PD	PVD
ED		1.00		1.00				
I-DT	7.00	5.00	8.26	9.00	1.13			
I-HMM	4.07	7.04	4.98	6.85	5.67			
I-KF	3.02	6.17	3.15	7.85	7.74			
I-MST	1.33	8.91	1.37	2.13	2.76			
I-VDT	6.28	4.00	4.07	3.93	5.07	2.00	1.78	2.00
I-VMT	2.15	2.13	2.43	3.04	6.67	1.00	3.00	1.00
I-VT	4.72	7.87	5.98	4.98	4.04			
I-VVT	7.52	2.87	7.09	6.22	2.91	3.00	1.22	3.00
BIT	8.91	10.00	7.67					
	$x^2$204.01 p<0.001	204.39	147.03	179.21	131.23	46.00	38.17	46.00

TABLE 14. Classification results of the two clusters.

CLUSTER	FD (ms)
I	54286.87
II	6314.26

TABLE 15. Classification results of the three clusters.

CLUSTER	FN	FD (ms)	SN	SD (ms)	SAD (deg)
I	15.17	6527.02	24.82	2261.56	4.39
II	12.17	9042.51	13.87	1599.52	5.18
III	11.75	3801.77	18.02	1367.94	5.04

The mean values obtained from the three clusters of eight algorithms are given in Table 15. The hierarchical clustering analysis showed that the three-cluster structure resulted from the differences in the methods and threshold values of the algorithms.

The algorithms I-DT, I-HMM, I-KF, I-MST, I-VDT, I-VMT, I-VT and I-VVT are included in a single software package (Appendix-A) that is open-source but password-protected. The combination of algorithms with different features in a single interface is very useful and provides great convenience. Researchers can easily use the algorithms for their data by making the necessary changes through the interface. This software also allows graphical representation of the results, thereby offering significant convenience for researchers that are not experienced in this field. Based on the results of this study, the following important points can be highlighted:

- The most common eye-movement event classified by the algorithms was FD, followed by FN, FD, SN, SD, and SAD.
- Different algorithms provided different classification results due to the methods and threshold values they use. Despite the significant effect of the threshold values on the classification results, there is no consensus in the literature concerning how to decide on the appropriate value for each algorithm.
- According to hierarchical clustering based on the mean FD values produced by all ten algorithms, two clusters were identified: Cluster-1 consisted of the BIT algorithm and Cluster-2 contained the remaining nine algorithms. The different results obtained from BIT may result from the ability of this algorithm to automatically determine the threshold values based on the provided eye data whereas the default threshold values should be defined by the researcher for the remaining algorithms.
- As BIT and ED are limited in their classification of eye-movement events, another hierarchical clustering analysis was performed by excluding these algorithms. The results of hierarchical clustering based on the means of FN, FD, SN, SD, and SAD produced by the remaining eight algorithms (I-DT, I-HMM, I-KF, I-MST, I-VDT, I-VMT, I-VT and I-VVT) revealed three clusters: Cluster-1 consisted of the I-DT algorithm, which is dispersion-based, and the I-VDT algorithm, a modified combination of the velocity-based I-VT and dispersion-based I-DT algorithms. Dispersion-based algorithms are based on the x and y coordinates of the eye data for classifying eye-movements into fixation and saccade events. The I-VDT algorithm differs from modified velocity-based algorithms (I-VVT and I-VMT) in that it separates smooth pursuits from fixations by employing a modified dispersion threshold identification method. Therefore, the results of I-VDT differed from the remaining modified algorithms (I-VVT and I-VMT) as similarly reported in previous research [31]. Cluster-2 comprised the I-HMM, I-KF, I-MST, and I-VT algorithms. Except for the I-MST algorithm, all the remaining algorithms in this cluster are velocity-based and use the velocity of eye movements to separate fixation from saccade events. The I-MST algorithm has the advantage of correctly identifying the anchor points if a large part of the signal is missing, a property which makes I-MST a highly flexible and controllable eye-movement detection tool [33]. In addition, this algorithm produces similar results to velocity-based algorithms probably due to the threshold values used but the reasons behind this should be further investigated. The algorithms I-VMT and I-VVT were included in Cluster-3, with the common property that they are both modified versions of the velocity-based algorithm I-VT. These two algorithms first utilize the velocity threshold for classifying saccades, then inves-

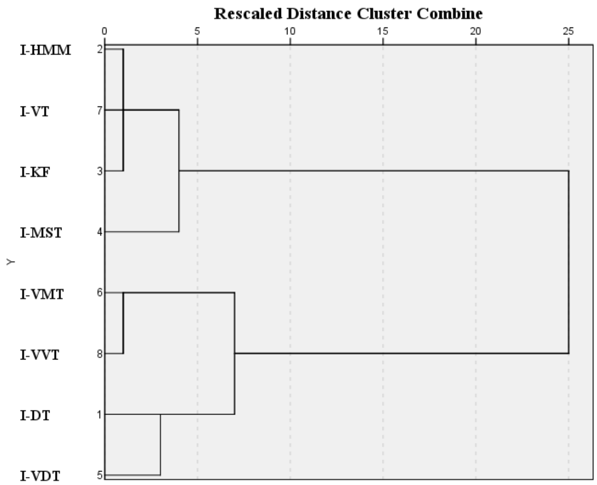

FIGURE 3. Dendrogram of eight algorithms using ward's linkage method.

tigate the eye-movement samples to distinguish smooth pursuits from fixations. As a result, both of these algorithms are more suitable for dynamic stimuli, such as video-viewing, where objects are moving [29], [31].

- This study experimentally evaluated open-source eye movement classification algorithms using the same dataset collected from the domain of neurosurgery. There are other studies in the literature that have investigated different eye movement classification algorithms [1], [29], [34] but none has performed a comparison of the ten algorithms used in this study.

V. CONCLUSION

In this study, the data was recorded and classified while surgeons performed four different surgical tasks under three hand conditions, which allowed a comprehensive comparison of the algorithms. Although these ten algorithms are widely used for classifying eye-movement events in different domains, there is no experimental study that compared the differences between them. In addition, the domain of the current study differs from previous research. To the best of our knowledge, no other study has been conducted with surgeons from the field of neurosurgery and ENT to evaluate eye-movement classification algorithms; thus, the present contributes to the literature in terms of how to decide on an appropriate algorithm for eye-movement studies undertaken in different domains.

The results reveal that, the threshold values critically affect the classification results of the algorithms. Additionally, dispersion-based and velocity-based algorithms provided different results. Accordingly, it can be concluded that if the threshold values are not specified, the use of the BIT algorithm is appropriate for classifying eye-movement events because this algorithm can individually specify the threshold values for the eye-movement data gathered using an eye-tracking device.

Furthermore, as it is a skill-based training and requires individualized training approaches, for the simulation-based surgical training programs, the use of individualised threshold values in the BIT algorithm can be more beneficial in classifying the raw eye data, and thus evaluating the individual progress levels of trainees based on their eye movement behaviours. In cases where the default threshold values are known, the remaining nine algorithms can also be applied. In conclusion, for a better interpretation of eye research data, the appropriateness of the algorithms for the specific requirements of research should be considered.

The results of this study provided several insights into the use of eye-movement event classification for training procedures in a surgical simulation environment. It is considered that these results can also be implemented for other domains that aim to investigate human behaviours through eye movements.

APPENDIX A

Sources of the Evaluated Algorithms

Binocular-Individual Threshold (BIT) Algorithm

The MATLAB source code of BIT algorithm can be downloaded from the webpage

(http://www.bm.ust.hk/mark/faculty-and-staff/directory/85).

An Adaptive Event Detection (ED) Algorithm

The source code of ED algorithm can be downloaded from authors' webpage

(http://www.humlab.lu.se/en/person/MarcusNystrom)

Other Algorithms

The algorithms I-DT, I-HMM, I-KF, I-MST, I-VDT, I-VMT, I-VT and I-VVT can be downloaded from (http://cs.txstate.edu/~ok11/emd_offline.html). However researchers are expected to send an e-mail to the author for explaining their research purpose and asking their permission. The author provides a software password and the algorithms become available for the research purposes.

APPENDIX B

Threshold Values of the Algorithms

	I-VT	I-VVT	I-VMT	I-DT	I-VDT
Saccade Detection Threshold	70°/s	70°/s	70°/s		70°/s
Fixation Detection Threshold		20°/s			
Temporary Window Length Threshold			0.5 s		
Range Threshold			0.1		
Dispersion Duration Threshold				100ms	100ms
Dispersion Threshold				1.35°	1.35°

	I-HMM	I-KF	I-MST	ED	BIT
Saccade Detection Threshold	70°/s				
Viterbi Sample Size Threshold	200				
Baum-Welch Reiteration Threshold	5				
Sampling Window Size Threshold		5			
Deviation Threshold		1000			
Chi Threshold		15			
Window Size Threshold			200		
Saccade Detection Threshold, deg			0.6°		
Minimum Saccade Duration Threshold				20ms	
Minimum Fixation Duration Threshold				40ms	

ACKNOWLEDGMENT

The authors would like to thank to the Hacettepe University Medical School for their valuable support throughout the research.

REFERENCES

[1] O. V. Komogortsev, D. V. Gobert, U. K. S. Jayarathna, D. H. Koh, and S. M. Gowda, "Standardization of automated analyses of oculomotor fixation and saccadic behaviors," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 11, pp. 2635–2645, Nov. 2010.

[2] G. G. M. Dalveren and N. E. Cagiltay, "Insights from surgeons' eye-movement data in a virtual simulation surgical training environment: Effect of experience level and hand conditions," *Behav. Inf. Technol.*, vol. 37, no. 5, pp. 517–537, 2018.

[3] Y.-T. Huang, "The female gaze: Content composition and slot position in personalized banner ads, and how they influence visual attention in online shoppers," *Comput. Hum. Behav.*, vol. 82, pp. 1–15, May 2018.

[4] J. Mou and D. Shin, "Effects of social popularity and time scarcity on online consumer behaviour regarding smart healthcare products: An eye-tracking approach," *Comput. Hum. Behav.*, vol. 78, pp. 74–89, Jan. 2018.

[5] H. Jarodzka, K. Holmqvist, and H. Gruber, "Eye tracking in educational science: Theoretical frameworks and research agendas," *J. Eye Movement Res.*, vol. 10, no. 1, pp. 1–8, 2017.

[6] H. Park, S. Lee, M. Lee, M.-S. Chang, and H.-W. Kwak, "Using eye movement data to infer human behavioral intentions," *Comput. Hum. Behav.*, vol. 63, pp. 796–804, Oct. 2016.

[7] A. Selivanova and P. F. Krabbe, "Eye tracking to explore attendance in health-state descriptions," *PLoS ONE*, vol. 13, no. 1, 2018, Art. no. e0190111.

[8] P. Li, X. Hou, X. Duan, H. Yip, G. Song, and Y. Liu, "Appearance-based gaze estimator for natural interaction control of surgical robots," *IEEE Access*, vol. 7, pp. 25095–25110, 2019.

[9] X. Zhang, S.-M. Yuan, M.-D. Chen, and X. Liu, "A complete system for analysis of video lecture based on eye tracking," *IEEE Access*, vol. 6, pp. 49056–49066, 2018.

[10] T. J. Crawford, S. Higham, T. Renvoize, J. Patel, M. Dale, A. Suriya, and S. Tetley, "Inhibitory control of saccadic eye movements and cognitive impairment in Alzheimer's disease," *Biol. Psychiatry*, vol. 57, no. 9, pp. 1052–1060, 2005.

[11] G. G. M. Dalveren, N. E. Çağiltay, E. Özçelik, and H. Maraş, "Simulation-based environments for surgical practice," in *Proc. 4th Int. Conf. Control, Decis. Inf. Technol.*, Apr. 2017, pp. 1153–1156.

[12] J. A. Sweeney, B. J. Brew, J. G. Keilp, J. J. Sidtis, and R. W. Price, "Pursuit eye movement dysfunction in HIV-1 seropositive individuals," *J. Psychiatry Neurosci.*, vol. 16, no. 5, p. 247, 1991.

[13] K.-M. Flechtner, B. Steinacher, R. Sauer, and A. Mackert, "Smooth pursuit eye movements in schizophrenia and affective disorder," *Psychol. Med.*, vol. 27, no. 6, pp. 1411–1419, 1997.

[14] M. S. Bolding, A. C. Lahti, D. White, C. Moore, D. Gurler, T. J. Gawne, and P. D. Gamlin, "Vergence eye movements in patients with schizophrenia," *Vis. Res.*, vol. 102, pp. 64–70, Sep. 2014.

[15] L. Larsson, M. Nystrom, and M. Stridh, "Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 9, pp. 2484–2493, Sep. 2013.

[16] P. M. van Leeuwen, S. de Groot, R. Happee, and J. C. de Winter, "Differences between racing and non-racing drivers: A simulator study using eye-tracking," *PLoS ONE*, vol. 12, no. 11, 2017, Art. no. e0186871.

[17] H. P. Bui, S. Tomar, and S. P. Bordas, "Corotational cut finite element method for real-time surgical simulation: Application to needle insertion simulation," *Comput. Methods Appl. Mech. Eng.*, vol. 345, pp. 183–211, Mar. 2019.

[18] H. P. Bui, S. Tomar, H. Courtecuisse, M. Audette, S. Cotin, and S. P. Bordas, "Controlling the error on target motion through real-time mesh adaptation: Applications to deep brain stimulation," *Int. J. Numer. Methods Biomed. Eng.*, vol. 34, no. 5, 2018, Art. no. e02958.

[19] H. P. Bui, S. Tomar, H. Courtecuisse, S. Cotin, and S. P. Bordas, "Real-time error control for surgical simulation," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 3, pp. 596–607, Mar. 2018.

[20] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. A. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Med. Image Anal.*, vol. 18, no. 2, pp. 394–410, 2014.

[21] P. Hauseux, J. S. Hale, S. Cotin, and S. P. Bordas, "Quantifying the uncertainty in a hyperelastic soft tissue model with stochastic parameters," *Appl. Math. Model.*, vol. 62, pp. 86–102, Oct. 2018.

[22] N. E. Cagiltay, E. Ozcelik, G. Sengul, and M. Berker, "Construct and face validity of the educational computer-based environment (ECE) assessment scenarios for basic endoneurosurgery skills," *Surgical Endoscopy*, vol. 31, no. 11, pp. 4485–4495, 2017.

[23] N. E. Cagiltay, "The effect of training, used-hand, and experience on endoscopic surgery skills in an educational computer-based simulation environment (ECE) for endoneurosurgery training," *Surgical Innov.*, Aug. 2019, doi: 10.1177/1553350619861563.

[24] G. G. M. Dalveren, N. E. Cagiltay, E. Ozcelik, and H. Maras, "Insights from pupil size to mental workload of surgical residents: Feasibility of an educational computer-based surgical simulation environment (ECE) considering the hand condition," *Surgical Innov.*, vol. 25, no. 6, pp. 616–624, 2018.

[25] D. D. Salvucci and J. R. Anderson, "Automated eye-movement protocol analysis," *Hum.-Comput. Interact.*, vol. 16, no. 1, pp. 39–86, 2001.

[26] G. G. M. Dalveren and N. E. Cagiltay, "Using eye-movement events to determine the mental workload of surgical residents," *J. Eye Movement Res.*, vol. 11, no. 4, p. 3, 2018.

[27] L. Lévêque, H. Bosmans, L. Cockmartin, and H. Liu, "State of the art: Eye-tracking studies in medical imaging," *IEEE Access*, vol. 6, pp. 37023–37034, 2018.

[28] D. A. Robinson, J. Gordon, and S. Gordon, "A model of the smooth pursuit eye movement system," *Biol. Cybern.*, vol. 55, no. 1, pp. 43–57, 1986.

[29] R. Andersson, L. Larsson, K. Holmqvist, M. Stridh, and M. Nyström, "One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms," *Behav. Res. Methods*, vol. 49, no. 2, pp. 616–637, 2017.

[30] E. Koronaki, P. A. Gkinis, L. Beex, S. P. Bordas, C. Theodoropoulos, and A. G. Boudouvis, "Classification of states and model order reduction of large scale Chemical Vapor Deposition processes with solution multiplicity," *Comput. Chem. Eng.*, vol. 121, pp. 148–157, Feb. 2019.

[31] O. V. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades," *Behav. Res. Methods*, vol. 45, no. 1, pp. 203–215, 2013.

[32] O. V. Komogortsev, S. Jayarathna, D. H. Koh, and S. M. Gowda, "Qualitative and quantitative scoring and evaluation of the eye movement classification algorithms," in *Proc. Symp. Eye-Tracking Res. Appl.*, 2010, pp. 65–68.

[33] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proc. Symp. Eye Tracking Res. Appl.*, 2000, pp. 71–78.

[34] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," *Behav. Res. Methods*, vol. 42, no. 1, pp. 188–204, 2010.

[35] R. van der Lans, M. Wedel, and R. Pieters, "Defining eye-fixation sequences across individuals and tasks: The binocular-individual threshold (BIT) algorithm," *Behav. Res. Methods*, vol. 43, no. 1, pp. 239–257, 2011.

[36] H. Kurdak, D. AltIntaŞ, and F. Doran, "Medical education in Turkey: Past to future," *Med. Teacher*, vol. 30, no. 8, pp. 768–773, 2008.

[37] E. Dalmaijer, "Is the low-cost EyeTribe eye tracker any good for research?" *PeerJ PrePrints*, vol. 2, p. e585v1, Nov. 2014.

[38] J. Coyne and C. Sibley, "Investigating the use of two low cost eye tracking systems for detecting pupillary response to changes in mental workload," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, 2016, vol. 60, no. 1, pp. 37–41.

[39] T. Bray, *The JavaScript Object Notation (JSON) Data Interchange Format*, document RFC 7159, 2017.

[40] E. McCrum-Gardner, "Which is the correct statistical test to use?" *Brit. J. Oral Maxillofacial Surg.*, vol. 46, no. 1, pp. 38–41, 2008.

GONCA GOKCE MENEKSE DALVEREN received the Ph.D. degree in software engineering from Atilim University. She is currently a Postdoctoral Researcher with the Department of Computer Science, Norwegian University of Science and Technology, and with the Information Systems Engineering Department, Atilim University. Her research interests include software engineering, eye tracking, medical informatics, and human–computer interaction.

NERGIZ ERCIL CAGILTAY received the degree in computer engineering and the Ph.D. degree in instructional technologies from Middle East Technical University, Turkey. Her main research interests include information systems, surgical education, medical information systems, instructional systems technologies, distance education, and e-learning.

• • •