© 2014 IJEDR | Volume 3, Issue 1 | ISSN: 2321-9939

Detected Eye Tracking Techniques: And Method Analysis Survey

[1]Ruchika A. Patel and [2]Mr. Sandip R. Panchal
[1]PG Student, [2]Assistant Professor
[1, 2]Department of Electronics & Communication
Sardar vallabhbhai Institute of technology Vasad, Gujarat, India

***Abstract* - Eye-gaze detection and tracking has been an active research field in the past years as it adds convenience to a variety of applications. It is considered a significant untraditional method of human computer interaction. A Motion analysis method is developed to track and detect eye blinking. They serve a wide range of severely disabled people who are left with minimal motor abilities. In this work we want to put some light on new system in which Using eye as an interface to communicate with system for people that are severely paralyzed or affected by diseases in which they are unable to move or control most of their body parts except for their eyes. This paper gives overview of different techniques and describes best possible methods of eye blink detection techniques. The main propose system is the motion analysis method and finding frame difference used for tracking intentionally blink of eyes.**

***Keywords* - Eye tracking, Eye detection Method, Eye Tracking Applications, Detection of basic eye movement types, Comparison of eye tracking methods**

I. INTRODUCTION

Eyes and their movements are important in expressing a person's desires, needs and emotional states [1]. The significance of eye movements with regards to the perception of and attention to the visual world is certainly acknowledged since it is the means by which the information needed to identify the characteristics of the visual world is gathered for processing in the human brain. Hence, robust eye detection and tracking are considered to play a crucial role in the development of human-computer interaction, creating attentive user interfaces and analyzing human affective states. Eye tracking detection is widely investigated as alternative interface methods. They are considered to be easier to use than other methods such as voice recognition or EEG/ECG signals. They also have achieved higher accuracy and performance. In addition, using eye tracking detection as alternative interface, control or communication methods is beneficial for a wide range of severely disabled people who are left with minimal ability to perform voluntary motion. There are many approaches introduced in literature focusing on eye tracking. They can be used as a base to develop an eye tracking system which achieves the highest accuracy, best performance and lowest cost. There are many proposed approaches. Some approaches may be implemented using low computational hardware such as a microcontroller due to the simplicity of the used algorithm. This paper presents a survey of different eye tracking techniques reported in the literature along with examples of various applications employing these technologies.

The rest of the paper is outlined as follows. Different methods of eye tracking are investigated in Section 2. Detection of basic eye movement types is described in section 3. An Experimental result of existing eye tracking Techniques is also described in section 4. Finally, Section 5 draws the conclusions.

II. EYE TRACKING

The geometric and motion characteristics of the eyes are unique which makes gaze estimation and tracking important for many applications such as human attention analysis, human emotional state analysis, interactive user interfaces and human factors. There are many different approaches for implementing eye detection and tracking systems [2]. Many eye tracking methods were presented in the literature. However, the research is still on-going to find robust eye detection and tracking methods to be used in a wide range of applications.

2.1 Sensor-based eye tracking (EOG)

Some eye tracking systems detect and analyze eye movements based on electric potentials measured with electrodes placed in the region around the eyes. This electric signal detected using two pairs of electrodes placed around one eye is known as electrooculogram (EOG). When the eyes are in their origin state, the electrodes measure a steady electric potential field. If the eyes move towards the periphery, the retina approaches one electrode and the cornea approaches the other. This changes the orientation of the dipole and results in a change in the measured EOG signal. Eye movement can be tracked by analyzing the changes in the EOG signal [3].

2.2 Computer-vision-based eye tracking

Most eye tracking methods presented in the literature use computer vision based techniques. In these methods, a camera is set to focus on one or both eyes and record the eye movement. The main focus of this paper is on computer vision based eye detection and gaze tracking. There are two main areas investigated in the field of computer vision based eye tracking. The first area considered is eye detection in the image, also known as eye localization. The second area is eye tracking, which is the process of eye gaze direction estimation. Based on the data obtained from processing and analyzing the detected eye region, the direction of eye gaze can be estimated then it is either used directly in the application or tracked over subsequent video frames in the case of real-time eye tracking systems. Eye detection and tracking is still a challenging task, as there are many issues associated with such systems. These issues include degree of eye openness, variability in eye size, head pose, etc. Different applications that use eye tracking are affected by these issues at different levels. Several computer-vision-based eye tracking approaches have been introduced.

Pattern recognition for eye tracking

Different pattern recognition techniques, such as template matching and classification, have proved effective in the field of eye tracking. Raudonis et al. [4] used principal component analysis (PCA) to find the first six principal components of the eye image to reduce dimensionality problems, which arise when using all image pixels to compare images. Then, Artificial Neural Network (ANN) is used to classify the pupil position. The training data for ANN is gathered during calibration where the user is required to observe five points indicating five different pupil positions. The use of classification slows the system and hence it requires some enhancements to be applicable. In addition, the system is not considered a real time eye tracking system. The proposed algorithm was not tested on a known database which means the quality of the system might be affected by changes in lighting conditions, shadows, distance of the camera, the exact position in which the camera is mounted, etc. The algorithm requires processing which cannot be performed by low computational hardware such as a microcontroller.

Tang and Zhang [5] suggested a method that uses the detection algorithm combined with gray prediction to serve eye tracking purposes. The GM(1,1) model is used in the prediction of the location of an eye in the next video frame. The predicted location is used as the reference for the region of eye to be searched. The method uses low-level data in the image in order to be fast but there are no experimental results evaluating the performance of the method.

Kuo et al. [6] proposed an eye tracking system that uses the Particle filter which estimates a sequence of hidden parameters depending on the data observed. After detecting possible eyes positions, the process of eye tracking starts. For effective and reliable eye tracking, the gray level histogram is selected as the characteristics of the particle filter. Using low-level features in the image makes it a fast algorithm. High accuracy is obtained from the system; however, the real time performance was not evaluated, the algorithm was tested on images not videos and the images were not taken from a known database and, thus, the accuracy and performance of the algorithm may decrease when utilized in a real-world application.

Lui et al. [7] suggested a fast and robust eye detection and tracking method which can be used with rotated facial images. The camera used by the system is not head mounted. A Viola-Jones face detector, which is based on Haar features, is used to locate the face in the whole image. Then, Template Matching (TM) is applied to detect eyes. Zernike Moments (ZM) is used to extract rotation invariant eye characteristics. Support Vector Machine (SVM) is used to classify the images to eye/non-eye patterns. The exact positions of the left and right eyes are determined by selecting the two positions having the highest values among the found local maximums in the eye probability map. Detecting the eye region is helpful as a pre-processing stage before iris/pupil tracking. Especially, it allows for eye detection in rotated facial images. This work presented a simple eye tracking algorithm but the results of the method evaluation were not reported and thus the proposed eye tracking method is weak and not usable.

Hotrakool et al. [8] introduced an eye tracking method based on gradient orientation pattern matching along with automatic template updates. The method detects the iris based on low level features and motion detection between subsequent video frames. The method can be used in applications that require real-time eye tracking with high robustness against change in lighting conditions during operation. The computational time is reduced by applying down-sampling on the video frames. The method achieves high accuracy. However, the experiments were performed on videos of a single eye, which eliminates all surrounding noise, and a known database was not used. The method detects the iris but does not classify its position. The method requires minimal CPU processing time among other real-time eye tracking methods investigated in this survey. The motion detection approach discussed in this paper is worth being considered in new algorithms to obtain the minimal required CPU processing time in eye tracking applications.

Yuan and Kebin [9] presented Local and Scale Integrated Feature (LoSIF) as a new descriptor for extracting the features of eye movement based on a non-intrusive system, which gives some tolerance to head movements. The feature descriptor uses two-level Haar wavelet transform, multi-resolution characteristics and effective dimension reduction algorithm, to find the local and scale eye movement features. Support Vector Regression is used in mapping between the appearance of the eyes and the gaze direction, which correspond to each eye's appearance. The focus of this method is to locate the eye without classifying its position or gaze direction. The method was found to achieve high accuracy in iris detection, which makes this method useful in iris detection and segmentation which is important for eye tracking. However, the real-time performance was not evaluated.

Fu and Yang [10] proposed a high-performance eye tracking algorithm in which two eye templates, one for each eye, are manually extracted from the first video frame for system calibration. The face region in a captured frame is detected and a normalized 2-D cross-correlation is performed for matching the template with the image. Eye gaze direction is estimated by iris detection using edge detection and Hough circle detection. They used their algorithm to implement a display control application. However, it has an inflexible calibration process. The algorithm was not tested on a variety of test subjects and the results were not clearly reported which requires the algorithm to be investigated carefully before choosing to implement it.

Mehrubeoglu et al. [11] introduced an eye detection and tracking system that detects the eyes using template matching. The system uses a special customized smart camera which is programmed to continuously track the user's eye movements until the user stops it. Once the eye is detected, the region of interest (ROI) containing only the eye is extracted with the aim of reducing the processed region. From their work, it can be concluded that it is a fast eye tracking algorithm with acceptable performance. The algorithm could be a nice feature to be added to modern cameras. A drawback is that the experiments were not performed using a database containing different test subjects and conditions, which reduces the reliability of the results. In addition, the algorithm locates the coordinates but does not classify the eye gaze direction to be left, right, up or down.

Eye tracking based on corneal reflection points

Many computer vision based eye trackers use light reflection points on the cornea to estimate the gaze direction. Figure 1 shows the corneal reflection points in an eye image [12]. Another name for eye images containing corneal reflection points is Purkinje Image. When using this approach, the vector between the center of the pupil and the corneal reflections is used to compute the gaze direction. A simple calibration procedure of the individual is usually needed before using the eye tracker [2_12].

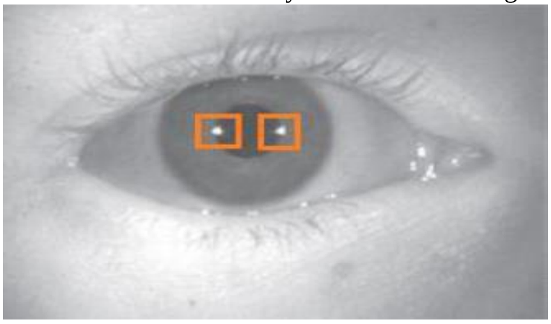


Figure1. Corneal reflection points [12].

Yang et al. [13] proposed a scheme which employs gray difference between the face, pupils and corneal reflection points for eye detection. The proposed scheme was tested under a cross-ratio-invariant-based eye tracking system. The test included users wearing glasses and other accessories and the results showed the ability of the system to eliminate the optical reflective effect of accessories and glasses. The scheme first prepares for gaze tracking by a preprocessing stage applied on cropped faces. This is particularly useful in applications which use a very close camera. The results are not detailed and not performed on a database containing various test subjects under different conditions which makes the algorithm weak when considered for use in real-world applications. In addition, the required CPU time was not addressed and thus the algorithm needs optimization to determine whether it works in real-time applications.

Yang et al. [14] presented an algorithm based on the difference in the gray level of the pupil region and the iris region. The image of the eye region is binarized and an estimate position of the pupil is detected. The exact position of the pupil is found by the vertical integral projection and the horizontal integral projection. The projection area contains the corneal glints and the pupil. The gray level of the pixels representing reflection points is the highest among all pixels. To apply their presented gaze tracking method, the points of corneal reflection must have known coordinates. The system uses hardware which adds limitations and inflexibility for pupil detection. The algorithm was tested on images containing a part of the face. It needs to be applied on different test subjects to prove the accuracy and performance of the algorithm. The required CPU time was not reported which does not make the algorithm feasible for adoption in real-time applications. However, the experiments showed that the algorithm achieves reasonably high accuracy.

Eye tracking based on shape

Another approach for eye detection and tracking is to find the location of the iris or the pupil based on their circular shape or using edge detection. Chen and Kubo [15] proposed a technique where a sequence of face detection and Gabor filters is used. The potential face regions in the image are detected based on skin color. Then, the eye candidate region is determined automatically using the geometric structure of the face. Four Gabor filters with different directions (0, $\pi/4$, $\pi/2$, $3\pi/4$) are applied to the eye candidate region. The pupil of the eye does not have directions and thus, it can be easily detected by combining the four responses of the four Gabor filters with a logical product. The system uses a camera which is not head mounted. The accuracy of the algorithm is not investigated and the required CPU time is not mentioned which does not make the algorithm preferable for real-world applications compared to other algorithms.

Kocejko et al. [16] proposed the Longest Line Detection (LLD) algorithm to detect the pupil position. This algorithm is based on the assumption that the pupil is arbitrary circular. The longest vertical and horizontal lines of the pupil are found. The center of the longest line among the vertical and horizontal lines is the pupil center. The proposed eye tracking system requires inflexible hardware which requires relatively difficult installation. The accuracy is not discussed and the performance of the system might be affected by changes in illumination, shadows, noise and other effects because the experiments were performed under special conditions and did not use a variety of test samples in different conditions.

Khairosfaizal and Nor'aini [17] presented a straightforward eye tracking system based on mathematical Circular Hough transform for eye detection applied to facial images. The first step is detecting the face region which is performed by an existing face detection method. Then the search for the eye is based on the circular shape of the eye in a two dimensional image. Their work added value to academic research but not to real-world applications.

Pranith and Srikanth [18] presented a method which detects the inner pupil boundary by using Circular Hough transformation whereas the outer iris boundary is detected by circular summation of intensity from the detected pupil center and radius. This algorithm can be used in an iris recognition system for iris localization because it is applied to cropped eye images. However, it needs further analysis to obtain its accuracy and real-time performance by applying it on a database containing variant images.

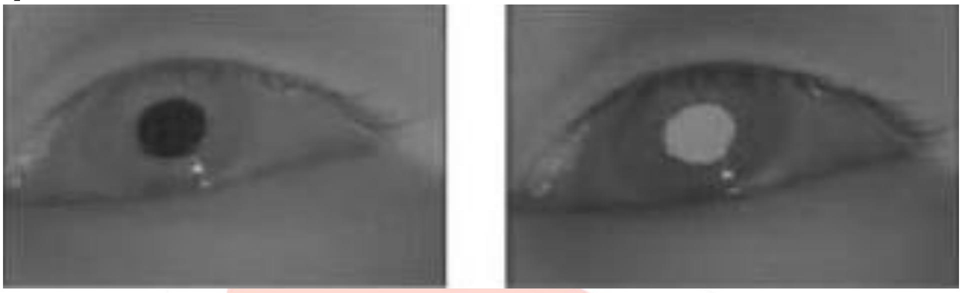
Sundaram et al. [19] proposed an iris localization method that identifies the outer and inner boundaries of the iris. The procedure includes two basic steps: detection of edge points and Circular Hough transform. Before these steps are applied, bounding boxes for iris region and pupil area are defined to reduce the complexity of the Hough transform as it uses a voting scheme that considers all edge points (Xi,Yi) over the image for all possible combinations of center coordinates (x0, y0) and different possible values of the radius r. The algorithm is applicable in applications where a camera close to the eye is used such as in iris recognition systems. The algorithm was tested on the UBIRIS database, which contains images with different characteristics like illumination change. The algorithm is relatively fast when compared to non real-time applications but it is not suitable for real-time eye tracking due to its complexity.

Alioua et al. [20] presented an algorithm that handles an important part of eye tracking, which is analyzing eye state (open/closed) using iris detection. Gradient image is used to mark the edge of the eye. Horizontal projection is then computed for the purpose of detecting upper and lower boundaries of the eye region and the Circular Hough transform is used for iris detection. This algorithm can be useful in eye blinking detection, especially since the possible output classes are limited: open, closed or not an eye. When used in a control application, it can increase the number of possible commands. Due to its importance, the algorithm is required to be fast but no required CPU time was mentioned, although indicating the required CPU time could be useful.

Eye tracking using dark and bright pupil effect

There are two illumination methods used in the literature for pupil detection: the dark pupil and the bright pupil method. In the dark pupil method, the location of a black pupil is determined in the eye image captured by the camera. This causes some issues when the user has dark brown eyes because of the low contrast between the brown iris and the black pupil. The bright pupil method uses the reflection of infrared light from the retina which makes the pupil appear white in the eye image. Figure 2 shows the dark and bright pupil effect.

Yoo et al. [21] proposed using cross-ratio-invariance for a gaze tracking algorithm. The proposed algorithm has been found to achieve very low accuracy. To enhance it, a virtual plane tangent to the cornea is added. The enhancement did not solve all issues as the system remains complex due to using two cameras to obtain the difference between dark and bright pupil images for pupil position detection [22].



(a) (b)

Figure 2 (a) Dark and (b) Bright pupil effects [2].

Eye tracking using eye models

Eye models can be used for eye tracking. Zhu and Ji [23] proposed two new schemes to allow natural head movement for eye tracking and minimize the calibration of the eye tracking system to only once for each new user. The first scheme estimates the 3-D eye gaze directly. The cornea of the eye is modeled as a convex mirror. Based on the characteristics of a convex mirror, the eye's 3-D optic axis is estimated. The visual axis represents the actual 3-D eye gaze direction of the user. It is determined after knowing the angle deviation between the visual axis and optic axis in calibration. The second scheme does not require estimating the 3-D eye gaze, and the gaze point can be determined implicitly using a gaze mapping function. The gaze mapping function is updated automatically upon head movement using a dynamic computational head compensation model. The proposed schemes require medical and physical background as well as image processing techniques. The schemes are complex and hence they are not applicable for real-time eye tracking. It would be beneficial if the used algorithms, equations and phases were optimized. In addition, the system uses inflexible hardware which is not preferred in real applications. The schemes were not tested on a database containing various test subjects and conditions.

Hybrid eye tracking techniques

A mix of different techniques can be used for eye tracking. Huang et al. [24] suggested an algorithm to detect eye pupil based on intensity, shape, and size. Special Infrared (IR) illumination is used and thus, eye pupils appear brighter than the rest of the face. The intensity of the eye pupil is used as the primary feature in pupil detection. However, some other bright objects might exist in the image. To separate the pupil from bright objects existing in the image, other pupil properties can be used, such as pupil size and shape. Support Vector Machine is used to locate the eye location from the detected candidates. The used hardware, including the IR LEDs and the IR camera, is not expensive. The algorithm has been used in a driver fatigue detection application. The algorithm can be considered a new beginning for real-time eye tracking systems if it is tested further with different test subjects and different classification functions in order to reach the most optimized eye algorithm. The required CPU time was not mentioned although it is important in driver fatigue detection applications as they are real-time applications.

Using a corneal reflection and energy controlled iterative curve fitting method for efficient pupil detection was proposed by Li andWee [12]. Ellipse fitting is needed to acquire the boundary of the pupil based on a learning algorithm developed to perform iterative ellipse fitting controlled by a gradient energy function. This method uses special hardware which has been implemented specifically for this algorithm. It has been used in a Field-of-View estimation application and can be used in other applications.

Coetzer and Hancke [25] proposed a system for eye tracking which uses an IR camera and IR LEDs. It captures the bright and dark pupil images subsequently such that they are effectively the same image but each has been taken in different illumination conditions. Two groups of infrared LEDs are synchronized with the IR camera. The first is placed close to the camera's lens to obtain the bright pupil effect and the second about 19.5 cm away from the lens, to produce the dark pupil effect. The images are then subtracted from each other and a binary image is produced by thresholding the difference. This technique has been presented by Hutchinson [26]. The binary image contains white blobs that are mapped to the original dark pupil image. The sub-images that result from the mapping are potential eye candidates. The possible eye candidate sub-images are classified into either eyes or non-eyes. Artificial neural networks (ANN), support vector machines (SVM) and adaptive boosting (AdaBoost) have been considered as classification techniques in this work. The system is ready for further improvements and enhancements. It was utilized in a driver fatigue monitoring system. It does not require calibration for each user because it uses a dataset for training and feature extraction. The best features to be used and flexible hardware implementation can be investigated further in order to make the algorithm a part of a bigger eye tracking system or eye location classification system. The images used in experiments were eye images after the background and noise were eliminated. This reduces the expectations of this algorithm performance in real applications.

III. DETECTION OF BASIC EYE MOVEMENT TYPES

Different types of eye movements can be detected from the processed EOG signals. In this work, saccades, fixations, and blinks form the basis of all eye movement features used for classification. The robustness of the algorithms for detecting these is key to achieving good recognition performance. Saccade detection is particularly important because fixation detection, eye movement encoding, and the wordbook analysis are all reliant on it. In the following, we introduce our saccade and blink detection algorithms and characterize their performance on EOG signals recorded under constrained conditions.

3.1 Saccade and Fixation Detection

For saccade detection, we developed the so-called Continuous Wavelet Transform—Saccade Detection (CWT-SD) algorithm (see Fig. 4 for an example). Inputs to CWT-SD are the denoised and baseline drift removed EOG signal components EOGh and EOGv. CWT-SD first computes the continuous 1D wavelet coefficients at scale 20 using a Haar mother wavelet. Let s be one of these signal components and the mother wavelet. The wavelet coefficient C_b^a of s at scale a and position b is defined

$$C_b^a\,(S) = \int_{IR} S(t)\,\frac{1}{\sqrt{a}}\,\varphi(\overline{\tfrac{t-b}{a}})dt$$

By applying an application-specific threshold th_{sd} on the coefficients $C_i(s) = C_i^{20}(s)$, CWT-SD creates a vector M with elements M_i:

$$M_i = \begin{cases} 1, & \forall_i: C_i(s) < -th_{sd} \\ -1, & \forall_i: C_i(s) > th_{sd} \\ 0, & \forall_i: -th_{sd} \le C_i(s) \le th_{sd} \end{cases}$$

This step divides EOGh and EOGv in saccadic (M = 1,-1) and nonsaccadic (fixational) (M = 0) segments. Saccadic segments shorter than 20 ms and longer than 200 ms are removed. These boundaries approximate the typical physiological saccade characteristics described in literature [30]. CWT-SD then calculates the amplitude and direction of each detected saccade. The saccade amplitude SA is the difference in EOG signal amplitude before and after the saccade (c.d. Fig. 4). The direction is derived from the sign of the corresponding elements in M. Finally, each saccade is encoded into a character representing the combination of amplitude and direction. For example, a small saccade in EOGh with negative direction gets encoded as “r” and a large saccade with positive direction as “L.”

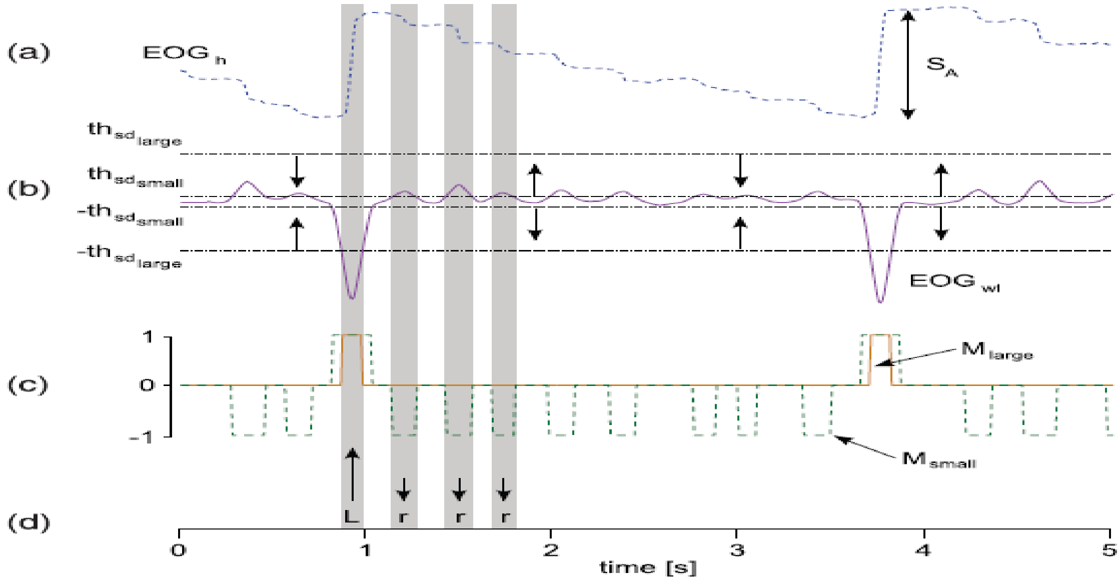


Figure 4.Continuous wavelet transform—saccade detection algorithm. (a) Denoised and baseline drift removed horizontal EOG signal during reading with example saccade amplitude (Sa); (b) The transformed wavelet signal (EOGWL), with application-specific small (th_{small}) and large (th_{large}) thresholds; (c) Marker vectors for distinguishing between small (m_{small}) and large (m_{large}) saccades; and (d) Example character encoding for part of the EOG signal.

Humans typically alternate between saccades and fixations. This allows us to also use CWT-SD for detecting fixations. The algorithm exploits the fact that gaze remains stable during a fixation. This results in the corresponding gaze points, i.e., The points in a visual scene that the gaze is directed at, to cluster together closely in time. Therefore, fixations can be identified by Thresholding on the dispersion of these gaze points [31]. For a segment S of length n comprised of a horizontal s_h and a vertical s_v EOG signal component, the dispersion is calculated as

$$\text{Dispersion}(S) = \max(s_h) - \min(s_h) + \max(s_v) - \min((s_v)$$

Initially, all nonsaccadic segments are assumed to contain a fixation. The algorithm then drops segments for which the dispersion is above a maximum threshold th_{fd} of 10,000 or if its duration is below a minimum threshold th_{fdt} of 200 ms. The value of th_{fd} was derived as part of the CWTSD evaluation; that of th_{fdt} approximates the typical average fixation duration reported earlier.

A particular activity may require saccadic eye movements of different distance and direction. For example, reading involves a fast sequence of small saccades while scanning each line of text, while large saccades are required to jump back to the beginning of the next line. We opted to detect saccades with two different amplitudes, "small" and "large."This requires two thresholds, th_{sd}small and th_{sd}large, to divide the range of possible values of C into three bands (see Fig. 4): no saccade (th_{sd}small < C < th_{sd}small), small saccade (th_{sd}large < C <th_{sd}small or th_{sd}small < C < th_{sd}large), and large saccade (C <th_{sd}large or C >th_{sd}large). Depending on its peak value, each saccade is then assigned to one of these bands. To evaluate the CWT-SD algorithm, we performed an experiment with five participants—one female and four males (age: 25-59 years, mean =36:8, sd =15:4). To cover effects of differences in electrode placement and skin contact, the experiment was performed on two different days; in between days the participants took off the EOG electrodes. A total of 20 recordings were made per participant, 10 per day. Each experiment involved tracking the participants' eyes while they followed a sequence of flashing dots on a computer screen. We used a fixed sequence to simplify labeling of individual saccades. The sequence was comprised of 10 eye movements consisting of five horizontal and eight vertical saccades. This produced a total of 591 horizontal and 855 vertical saccades.

3.2 Blink Detection

For blink detection, we developed the Continuous Wavelet Transform—Blink Detection (CWT-BD) algorithm. Similarly to CWT-SD, the algorithm uses a threshold th_{bd} on the wavelet coefficients to detect blinks in EOGv. In contrast to a saccade, a blink is characterized by a sequence of two large peaks in the coefficient vector directly following each other: one positive, the other negative. The time between these peaks is smaller than the minimum time between two successive saccades rapidly performed in opposite direction. This is because, typically, two saccades have at least a short fixation in between them. For this reason, blinks can be detected by applying a maximum threshold th_{bdt} on this time difference. We evaluated our algorithm on EOG signals recorded in a stationary setting from five participants looking at different pictures (two females and three males, age: 25-29 years, mean = 26:4, sd = 1:7). We labeled a total of 706 blinks by visual inspection of the vertical EOG signal component. With an average blink rate of 12 blinks per minute, this corresponds to about one hour of eye movement data. We evaluated CWTBD over sweeps of its two main parameters: th_{bd} = 100. . . 50;000 (in 500 steps) and th_{bdt} = 100. . . 1;000 ms (in 10 steps). The F1 score was calculated by matching blink events with the annotated ground truth. Time differences outside this range, as exemplarily shown for 300 and 1,000 ms, are already subject to a considerable drop in performance. This finding nicely reflects the values for the average blink duration cited earlier from the literature.

IV. EXPERIMENTAL RESULTS OF EXISTING EYE TRACKING TECHNIQUES

The performance of eye tracking and head movement detection systems is evaluated in terms of accuracy and required CPU processing time. This section compares the results of the methods described earlier in the survey.

4.1 Comparison of eye tracking methods

Table 1 Comparison of Eye Tracking Methods

Method	Detection Accuracy (%)	Angle Accuracy (degree)	CPU time (ms)
Eye tracking using pattern Recognition			
Randonis et al.[4]	100%	N/A	N/A
Kuo et al.[6]	90%	N/A	N/A
Yuan and Kebin[9]	N/A	1	N/A
Lui and Lui[7]	94.1%	N/A	N/A
Khairosfaizal and Nor'aini[17]	86%	N/A	N/A
Hotrakool et al.[8]	100%	N/A	12.92
Shape-based eye tracking			
Yang et al.[13]	N/A	0.5	N/A
Yang et al.[14]	N/A	Horizontal:0.327 Vertical:0.3	N/A
Mehrubeoglu et al.[11]	90%	N/A	49.7
Eye tracking using eye models			
Zhu and Ji [23] (First scheme)	N/A	Horizontal:1.14 Vertical:1.58	N/A
Zhu and Ji [23] (Second scheme)	N/A	Horizontal:0.327 Vertical:0.83	N/A

Eye tracking using hybrid techiques			
Li and wee[12]	N/A	0.5	N/A
Huang et al.[24]	95.63%	N/A	N/A
Coetzer and Hancke[25]	98.1%	N/A	N/A

4.2 Performance of eye tracking methods

Table 1 compares the described eye tracking methods in terms of eye detection accuracy, gaze angle accuracy and required CPU time.

V. CONCLUSION

Eye tracking is considered effective and reliable human-computer interaction and communication alternative methods. Hence, they have been the subject of many research works. Many approaches for implementing these technologies have been reported in the literature. This paper investigated existing methods of eye tracking. Many applications can benefit from utilizing effective eye tracking methods. However, the research is still facing challenges in presenting robust methods which can be used in applications to detect and track eye accurately. Eye tracking methods rarely investigate the required CPU time. However, real-time application requires investigating and optimizing the performance requirements. In addition, most studies do not test eye tracking using a known image database that contains variant images of different subjects in different conditions such as lighting conditions, noise, distances, etc. This makes the reported accuracy of a method less reliable because it may be affected by different test conditions.

REFERENCES

[1] T. Ohno, ``One-point calibration gaze tracking method,'' in *Proc. Symp. Eye Track. Res. Appl.*, 2006, p. 34.
[2] D. W. Hansen and Q. Ji, ``In the eye of the beholder: A survey of models for eyes and gaze,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478_500, Mar. 2010.
[3] J. Gips, P. DiMattia, F. X. Curran, and P. Olivieri, ``Using EagleEyes_An electrodes based device for controlling the computer with your eyes to help people with special needs,'' in *Proc. Interdisciplinary Aspects Comput. Help. People Special Needs*, 1996, pp. 77_84.
[4] V. Raudonis, R. Simutis, and G. Narvydas, ``Discrete eye tracking for medical applications,'' in *Proc. 2nd ISABEL*, 2009, pp. 1_6.
[5] J. Tang and J. Zhang, ``Eye tracking based on grey prediction,'' in *Proc. 1st Int. Workshop Educ. Technol. Comput. Sci.*, 2009, pp. 861_864.
[6] Y. Kuo, J. Lee, and S. Kao, ``Eye tracking in visible environment,'' in *Proc. 5th Int. Conf. IIH-MSP*, 2009, pp. 114_117.
[7] H. Liu and Q. Liu, ``Robust real-time eye detection and tracking for rotated facial images under complex conditions,'' in *Proc. 6th ICNC*, vol. 4. 2010, vpp. 2028_2034.
[8] W. Hotrakool, P. Siritanawan, and T. Kondo, ``A real-time eye-tracking method using time-varying gradient orientation patterns,'' in *Proc. Int. Conf. Electr. Eng., Electron. Comput. Telecommun. Inf. Technol.*, 2010, pp. 492_496.
[9] Z. Yuan and J. Kebin, ``A local and scale integrated feature descriptor in eye-gaze tracking,'' in *Proc. 4th Int. CISP*, vol. 1. 2011, pp. 465_468.
[10] B. Fu and R. Yang, ``Display control based on eye gaze estimation,'' in *Proc. 4th Int. CISP*, vol. 1. 2011, pp. 399_403.
[11] M. Mehrubeoglu, L. M. Pham, H. T. Le, R. Muddu, and D. Ryu, ``Realtime eye tracking using a smart camera,'' in *Proc. AIPR Workshop*, 2011, pp. 1_7.
[12] X. Li andW. G.Wee, ``An ef_cient method for eye tracking and eye-gazed FOV estimation,'' in *Proc. 16th IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 2597_2600.
[13] C. Yang, J. Sun, J. Liu, X. Yang, D. Wang, and W. Liu, ``A gray difference-based pre-processing for gaze tracking,'' in *Proc. IEEE 10th ICSP*, Oct. 2010, pp. 1293_1296.
[14] X.Yang, J. Sun, J. Liu, J. Chu,W. Liu, andY. Gao, ``Agaze tracking scheme for eye-based intelligent control,'' in *Proc. 8th WCICA*, 2010, pp. 50_55.
[15] Y. W. Chen and K. Kubo, ``A robust eye detection and tracking technique using gabor _lters,'' in *Proc. 3rd Int. Conf. IEEE Intell. Inf. Hiding Multi-media Signal Process.*, vol. 1. Nov. 2007, pp. 109_112.
[16] T. Kocejko, A. Bujnowski, and J. Wtorek, ``Eye mouse for disabled,'' in *Proc. IEEE Conf. Human Syst. Interact.*, May 2008, pp. 199_202.
[17] W. Khairosfaizal and A. Nor'aini, ``Eye detection in facial images using circular Hough transform,'' in *Proc. 5th CSPA*, Mar. 2009, pp. 238_242.
[18] A. Pranith and C. R. Srikanth, ``Iris recognition using corner detection,'' in *Proc. 2nd ICISE*, 2010, pp. 2151_2154.
[19] R. M. Sundaram, B. C. Dhara, and B. Chanda, ``A fast method for iris Localization,'' in *Proc. 2nd Int. Conf. EAIT*, 2011, pp. 89_92.
[20] N. Alioua, A. Amine, M. Rziza, and D. Aboutajdine, ``Eye state analysis using iris detection based on circular hough transform,'' in *Proc. ICMCS*, Apr. 2011, pp. 1_5.
[21] D. H. Yoo, J. H. Kim, B. R. Lee, and M. J. Chung, ``Non-contact eye gaze tracking system by mapping of corneal re_ections,'' in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 94_99.
[22] D. H. Yoo and M. J. Chung, ``A novel nonintrusive eye gaze estimation using cross-ratio under large head motion,'' *Special Issue on Eye Detection and Tracking, Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 25_51, 2005.
[23] Z. Zhu and Q. Ji, ``Novel eye gaze tracking techniques under natural head movement,'' *IEEE Trans. Biomed. Eng.*, vol. 54, no. 12, pp. 2246_2260, Dec. 2007.

[24] H. Huang, Y. S. Zhou, F. Zhang, and F. C. Liu, ``An optimized eye locating and tracking system for driver fatigue monitoring," in *Proc. ICWAPR*, vol. 3. 2007, pp. 1144_1149.
[25] R. C. Coetzer and G. P. Hancke, ``Eye detection for a real-time vehicle driver fatigue monitoring system," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2011, pp. 66_71.
[26] R. Lupu, R. Bozomitu, F. Ungureanu, and V. Cehan, ``Eye tracking based communication system for patient with major neuro-locomotor disabilities," in *Proc. IEEE 15th ICSTCC*, Oct. 2011, pp. 1_5.
[27] C. Calvi, M. Porta, and D. Sacchi, ``e5Learning, an e-learning environment based on eye tracking," in *Proc. 8th IEEE ICALT*, Jul. 2008, pp. 376_380.
[28] M. Porta, S. Ricotti, and C. J. Pere, ``Emotional e-learning through eye tracking," in *Proc. IEEE Global Eng. Educ. Conf.*, Apr. 2012, pp. 1_6.
[29] M. S. Devi and P. R. Bajaj, ``Driver fatigue detection based on eye tracking," in *Proc. 1st Int. Conf. Emerg. Trends Eng. Technol.*, 2008, pp. 649_652.
[30] A.T. Duchowski, Eye Tracking Methodology: Theory and Practice. Springer-Verlag New York, Inc., 2007.
[31] D.L. Donoho, "De-Noising by Soft-Thresholding," IEEE Trans. Information Theory, vol. 41, no. 3, pp. 613-627, May 1995.