



# A review on visible-light eye-tracking methods based on a low-cost camera

Alberto J. Molina-Cantero<sup>1</sup> · Clara Lebrato-Vázquez<sup>1</sup> · Juan A. Castro-García<sup>1</sup> · Manuel Merino-Monge<sup>1</sup> · Félix Biscarri-Triviño<sup>1</sup> · José I. Escudero-Fombuena<sup>1</sup>

Received: 22 February 2023 / Accepted: 24 January 2024  
© The Author(s) 2024

## Abstract

This paper is the first of a two-part study aiming at building a low-cost visible-light eye tracker (ET) for people with amyotrophic lateral sclerosis (ALS). The whole study comprises several phases: (1) analysis of the scientific literature, (2) selection of the studies that better fit the main goal, (3) building the ET, and (4) testing with final users. This document basically contains the two first phases, in which more than 500 studies, from different scientific databases (IEEE Xplore, Scopus, SpringerLink, etc.), fulfilled the inclusion criteria, and were analyzed following the guidelines of a scoping review. Two researchers screened the searching results and selected 44 studies ( $\kappa = 0.86$ , Kappa Statistic). Three main methods (appearance-, feature- or model- based) were identified for visible-light ETs, but none significantly outperformed the others according to the reported accuracy  $-p = 0.14$ , Kruskal–Wallis test (KW)-. The feature-based method is abundant in the literature, although the number of appearance-based studies is increasing due to the use of deep learning techniques. Head movements worsen the accuracy in ETs, and only a very few numbers of studies considered the use of algorithms to correct the head pose. Even though head movements seem not to be a big issue for people with ALS, some slight head movements might be enough to worsen the ET accuracy. For this reason, only studies that did not constrain the head movements with a chinrest were considered. Five studies fulfilled the selection criteria with accuracies less than 2°, and one of them is illumination invariant.

**Keywords** Eye tracker · Webcam · ALS

## 1 Introduction

Some diseases, like amyotrophic lateral sclerosis (ALS) or locked-in syndrome (LIS), confine people to a state in which it makes it hard for them to perform the simplest movement apart from blinking or moving the eyes. Several augmentative and alternative communication (AAC) solutions exist to facilitate the human-computer interaction (HCI) in these cases, of which, the ET is especially appropriate for people with good eye gaze control. ETs are mainly based

on the use of cameras adapted to the type of scene illumination: infrared (IR) or visible light. Many ETs detect the position of the IR beam reflected by the surface of the eye. The camera tracks the pupil and the shiny IR spot in the eye (acting as reference point), so the relative position between them determines the eye gaze. An initial calibration process allows obtaining a mathematical function, or relationship, between the eye gaze (input variable) and the coordinates on a computer screen (output). The camera can be placed close to the eye (the eye camera), attached to a helmet through an articulated rod, to accurately capture the eye movements (Pasarica et al. 2016), or below the computer screen, face camera, embedded in a structure that can also host the IR emitters (Kim et al. 2014). Both camera placements are sensitive to head movements, so frequent calibrations are needed to correct the loss in accuracy. To avoid this problem, some authors have included a second camera, or field camera, placed on the helmet, that records the scene the subject

---

Alberto J. Molina-Cantero, Clara Lebrato-Vázquez, Juan A. Castro-García, Manuel Merino-Monge, Félix Biscarri-Triviño and José I. Escudero-Fombuena have contributed equally to this work.

---

✉ Alberto J. Molina-Cantero  
almolina@us.es

<sup>1</sup> Departamento de Tecnología Electrónica. ETS. Ingeniería Informática, Universidad de Sevilla, Av. Reina Mercedes sn, Seville 41020, Spain

sees. The ET software is then able to correct the eye-gaze according to the head position (Parada et al. 2015).

The main disadvantage of the IR-based ETs is the cost (Sharma and Jobanputra 2014), especially when several cameras and the helmet that supports them are needed. Nevertheless, low-cost and open-source solutions have emerged for eye tracking ([http://wiki.cogain.org/index.php/Eye\\_Trackers](http://wiki.cogain.org/index.php/Eye_Trackers)) with positioning accuracy similar to the proprietary counterparts (Dalmaijer 2014). The big inconvenience of open-source solutions, when they include the montage of hardware pieces, is that the setup may become difficult for people without the required knowledge, which severely limits their use.

Low-cost solutions would potentially allow ‘democratizing’ this technology. In this sense, an ET that only needed a webcam, which is usually included in most laptops and tablets, and based exclusively on an application program might result very useful, even though its accuracy was not as good as that achieved by other more complex and expensive ETs. Since many people are used to installing software, and there is no additional hardware to configure, this kind of solution may be very appealing for users, familiars, and caregivers. It has also been demonstrated that ETs based on a webcam have shown an accuracy similar to IR-based ETs when reasonably sized images are used, and the periphery of the screen is avoided (Burton et al. 2014).

This study looks for finding the best technologies and methods to build a visible-light low-cost eye tracker. To do that we need to find the answers to some research questions: The main one (RQ1) concern to the ET technology itself, its main features, the most employed algorithms, and their advantages and drawbacks. This question needs for a methodological review that was performed by applying the guidelines described in Tricco et al. (2018). Secondary questions are: Is there some method that outperforms or stands out from the rest? (RQ2); Which are the criteria to follow for the selection of the best webcam-based eye trackers? (RQ3). Sections 2 and 3 contain the literature review and the procedures followed to select the studies, while Sects. 4 and 5 contain the discussion and the conclusions respectively.

## 2 Literature review

### 2.1 Searching methodology

To answer RQ1, we performed a search in the following databases: IEEE Xplore, ScienceDirect, SpringerLink, ACM and Scopus. In general, the search covered meta-data, title and abstract containing some of the following keywords: ((webcam OR “low-cost camera” OR “consumer-grade camera”) AND (eye OR gaze) AND (tracker OR tracking) AND (HCI OR “human-computer interaction”). The search was

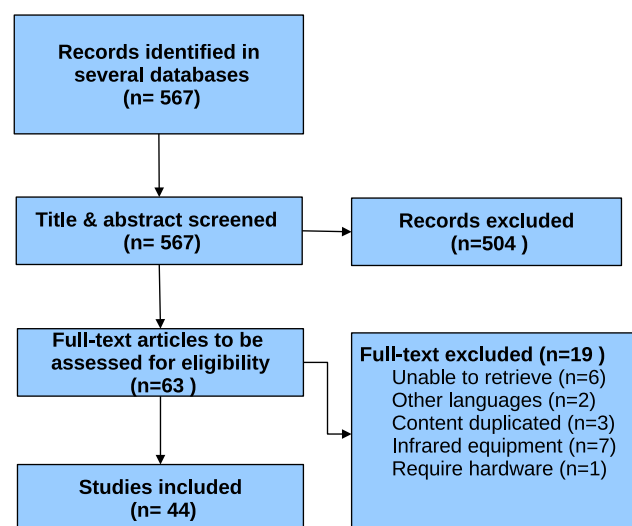
performed in September 2022 and included documents since 2005. After removing duplicates, an overall of 567 manuscripts satisfied the inclusion criteria. Searching phases and their results were sketched in the flowchart shown in Fig. 1.

Secondly, two researchers screened the manuscripts during an initial reviewing process that comprised the title and abstract, filtering out those that did not match with this study. To quantify the degree of agreement between the two researchers, the kappa statistic (Carletta 1996) was also obtained. Manuscripts that both researchers did not consider appropriated for this study were excluded. We found 44 documents fulfilling the reviewers’ criteria simultaneously, with a-value of 0.86, which means that the agreement level between them was ‘very good’ (Ashby 1991).

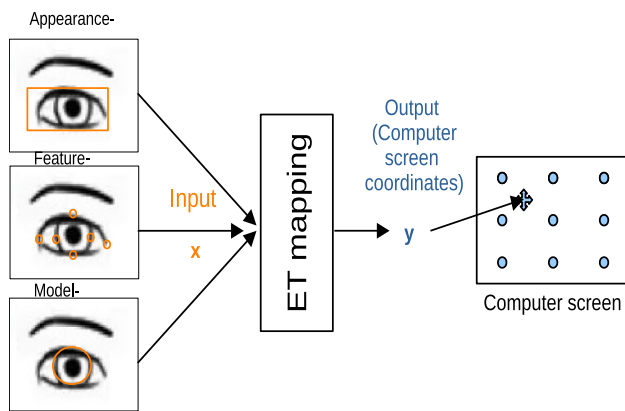
### 2.2 Eye tracking algorithms

The ET algorithm works like a mathematical function that maps an input independent variable,  $x$  (i.e., a feature in the eye images) to a vector,  $y$ , with the gaze position coordinates on the screen (Fig. 2). The algorithm must learn the relationship between the input and output vectors during the calibration process. According to Ferhat and Vilarifo (2016), three main types of visible light gaze estimation techniques can be found:

- Appearance-based. These methods only use the eye image pixel intensities. The estimation is based on creating a mapping between the eye gaze and a vector (feature) representation of the image via raster scanning.
- Feature-based. They are also based on using a mapping to estimate the eye gaze, nevertheless, here, a richer set



**Fig. 1** Flowchart showing the procedure followed, and the results obtained in each step



**Fig. 2** Graphical description of the three main methods found for ET algorithms

of features (not only pixels intensities) are used, such as the iris center, the canthus positions, some points of the eyelid border, etc.

- (c) **Model-based.** In this method, the gaze is estimated geometrically based on using model parameters. It is also called the geometric model.

The visible light eye tracker software algorithm comprises different steps:

1. **Eye location.** The first task is to extract the eye area -region of interest (ROI)- from the video input stream. Here, different algorithms can be used at spatial (like Viola-Jones) or temporal dimensions (to keep tracking the ROI and reduce computational time).
2. **Feature extraction or model fitting.** The main goal of this step is to prepare the input,  $x$ , for gaze estimation. The specific operations to accomplish depend on the type of the eye tracking technique:
  - (a) **Appearance- or feature-based eye trackers.** The vector  $x$  is the image itself.
  - (b) **Model-based trackers.** They search for the best fitting between the ROI and 2D- or 3D- eye and/or contour models to estimate the eye position and orientation.
3. **Gaze estimation.** For model-based trackers, the coordinates  $y$  are obtained by intersecting the eye gaze line (derived from the eye model) with the computer screen. The previous estimation of some parameters, such as the user-computer distance, is critical to ensure the correct operation. In general, ETs need to set a mapping between the output  $y$  and the input  $x$  vectors from  $N = (6, 9, 12, \dots)$  (Karamchandani et al. 2015) known screen positions. This can be accomplished by linear regression

or using more sophisticated techniques such as correntropy (Yang et al. 2019) or general regression neural network (GRNN) (Torricelli et al. 2008).

### 2.3 Full-text analysis

Tables 1, 2 and 3 detail the relevant information from the selected studies. The two first columns show the type of the ET and the steps followed in detecting the eye gaze. They therefore pinpoint the algorithms used in each stage of the processing pipeline (Fig. 3). Some of the selected studies were only focused on a part of the processing pipeline and did not develop a ET itself (Skodras and Fakotakis 2015; Xiao et al. 2018; Vater and León 2016).

• **Step 1: Eye location** It is the first step in every eye tracking algorithm. Several approaches can be found to deal with it. One of the most applied algorithms found in the literature, covering more than one half of the ones collected in this work, is based on the VJD detector (Viola and Jones 2001). Alternatively, in Zheng and Usagawa (2018), authors proposed the head detection with LBPs and the use of geometrical rules to locate the eyes. The LBP operator processes a  $3 \times 3$  pixel region by comparing its central pixel with its neighbors, setting them to 1 when their gray level is greater than the central pixel, or to 0 otherwise. The result is then arranged to build an 8-bit binary sequence. The face image is divided into  $N$  regions that are processed using the LBP transform and the LBP histogram. The  $N$  histograms are stitched into a new histogram creating a LBP feature that serves for training the head detection through the AdaBoost algorithm. A comparison between Haar-like features Viola-Jones detector (VJD) and LBP has been accomplished by Guennouni et al. (2015), showing that latter is computationally faster than Haar-like features with similar accuracy.

Other authors used facial landmark (FLs) (Xiao et al. 2018; Wojke et al. 2016; Solska and Kocejko 2022) to locate the eye position. This method usually needs for the VJD detector to find the head and delimit the region containing the face. As a first approach, in Anwar et al. (2018) utilized the active shape model (ASM) (Cootes and Taylor 1992), which derives from point distribution model (PDM), a method that defines a non-rigid object's contour from a collection of images including annotated points of interest. Scaling the principal components of the deformable model makes it possible its adaptation to the border of the object in a new image. In active appearance model (AAM) (Cootes et al. 2001; Wu et al. 2014) which can be considered as an extension of the ASM, both the grayscale image and the face shape are included in the model. constrained local model (CLM) (Asthana et al. 2014; Kim et al. 2016) utilizes an ensemble of local detectors, applied to a set of image patches cropped around the landmarks points, to find the optimal feature locations and maximize the responses of

**Table 1** Summarized description of the studies included in this work sorted by category

Method category	Step		Test	HP	II	Out.	Acc.	Reference	Comments
	1	2							
Appearance-based	TM, AR	Dif	-	X	↔	X	92–95%	Magee et al. (2008)	Only horizontal eye movements by subtracting left-right eye images
	VJD FL	KLD HoG	CoEn	X	-	↗	0.1°	Yang et al. (2019)	Dimensional reduction with KLD. Accuracy with 9 calibration points
	VID, Lum	PCA, IPROJ	Ip, LL	X	-	↗	0.8–1.1°	Liu et al. (2016, 2019)	Averaging of 5 X 7 calibration images (chin rest). Studless cropping effect on error
	FL, SCMS	PCA	SVR	X	-	↗	5.4°*	Mansanet et al. (2013)	Several regression algorithms were tested and two cameras. Kinect was worse because of its low resolution
	VID	TM	NN	X	-	↗	3°	Sewell and Komogortsev (2010)	12 X 11 ROI for a 2-layer NN
	FL	GP-LVM	GP	X	X	↗	1.9°	Wojke et al. (2016)	Camera and screen at different distances
	VJD	-	CNN	≈	X	⊞	85.6%	Modi and Singh (2022)	Algorithm to estimate user-computer distance based on facial features. 4 X 4 grid
	VJD	VJD	CNN	X	X	⊞	88.6%	Ansari et al. (2021)	5 X 4 grid
	-	FL	CNN	X	X	⊞	99%	Roy and Chanda (2022)	It detects up to 9 eye gestures
	FL	FL	CNN	X	✓	⊞	3.18°	Huang et al. (2021)	Eye states detecting by HoG and SVM. Head pose estimation to enable mouse operations like scroll up/down. 3x4 grid for calibration
Feature-based	VJD, LK	TM, HV-HST	GP	X	X	↗	1–5°	Ferhat et al. (2015)	Tested under natural lighting. Based on Ferhat et al. (2014)
	Dif	HT, CD	GRNN	≈	✓	↗	3.2°*	Torriceili et al. (2008)	15 gaze directions in a 3 X 5 grid. Blink detection. Multi-layer perceptron was also tested
	VJD	CoM	-	X	-	⊞	-	Rondio et al. (2012)	Only vertical and horizontal movements are tested
	LBP, GR	CoM	Ip	-	-	⊞	78–94%	Zheng and Usagawa (2018)	Accuracy depends on grid size (5 X 5, 3 X 3). Fast face detection
	VJD	HT, CD	Thr	-	-	↔	95%	Alva et al. (2017)	Detection of horizontal movements
	VJD	GF, BoI	SVM	-	-	⊞	97.8–92.97%	Lee et al. (2011)	Images are resized with a 2:1 ratio. 4 eye gaze directions and 3x3 grid
	Cr	Cr, Lum	-	X	✓	⊞	100%	Kourkoutsis et al. (2007)	Empirical window size. Accuracy for distance <3 pixels. Detection of some eye gazes
	FL	Lum, CoM	Thr	X	X	⊞	93–99%	Ahmed et al. (2019)	3 X 3 grid and 4 illumination conditions. Accuracy drops to 9% under poor illumination. Selection by blinks
	VJD	Orsu, CoM	Thr	X	X	⊞	-	Karamchandani et al. (2015)	5 calibration points to delimit a 4 X 4 grid

From left to right, the table contains the main algorithms used in each Step of the processing, how the proposal was tested (Test), if it is head (HP) or illuminance invariant (II), the type of Output, its Accuracy, the references (Ref) and some additional comments

Some acronyms: Step 1: active appearance model (AAM), antropometric rules (AR), active shape model (ASM), binarization of iris (BoI), constrained local model (CLM), chrominance component (Chr), difference between frames (Dif), facial landmark (FL), GEM, geometric rules (GR), LBP, luminance (Lum), Lucas-Kanade (LK), subspace constrained mean-shifts (SCMS), template matching (TM), Viola-Jones detector (VJD);

Step2: binarization of iris (BoI), between the eye feature (BTE), circle fitting (CF), corner detection (CD), convolutional neural network (CNN), center of mass (CoM), chrominance component (Chr), difference between frames (Dif), ellipse fitting (EFit), bidimensional fast Fourier transform (FFT2), FL, Gabor filter (GF), gaussian process latent variable model (GP-LVM), geometric rules (GR), gradient method (Grad), Hough transform (HT), histogram of oriented gradient (HoG) horizontal-vertical histogram (HV-HST), iterative circle fitting (ICF), integral projection (I PROJ), isocenter from isophote's curvature (IsoC), Kullback-Leibler divergence (KLD), luminance (Lum), Mean Shift (MS), normalized flow of luminance gradient (NFLG), principal component analysis (PCA), radial simetry (RS), ratio of sclera white pixels (RSWP) skewness distribution (Skw), snakusule (SNK), template matching (TM), Viola-Jones detector (VJD)

Step3: convolutional neural network (CNN), correntropy (CoEn), gaussian process regression (GP), general regression neural network (GRNN), iris color-luminance Model (ICLM), interpo- lation (Ip), linear homographic mapping (LHM), local learning (LL), neural network (NN), random forest algorithm (RFA), support vector machine (SVM), support vector regression (SVR), threshold (Thr);

Test:: dataset (DS), frame (Frm), people with disabilities (PWD), simulation (Sim), subjects (Sub), video (Vid);

Output: Point on screen (↗), box selection in a grid (⊞), joystick mode (⊞), left and right direction (↔) \* Accuracy obtained by Technology (2015)

Table 2 Continued from Table 1

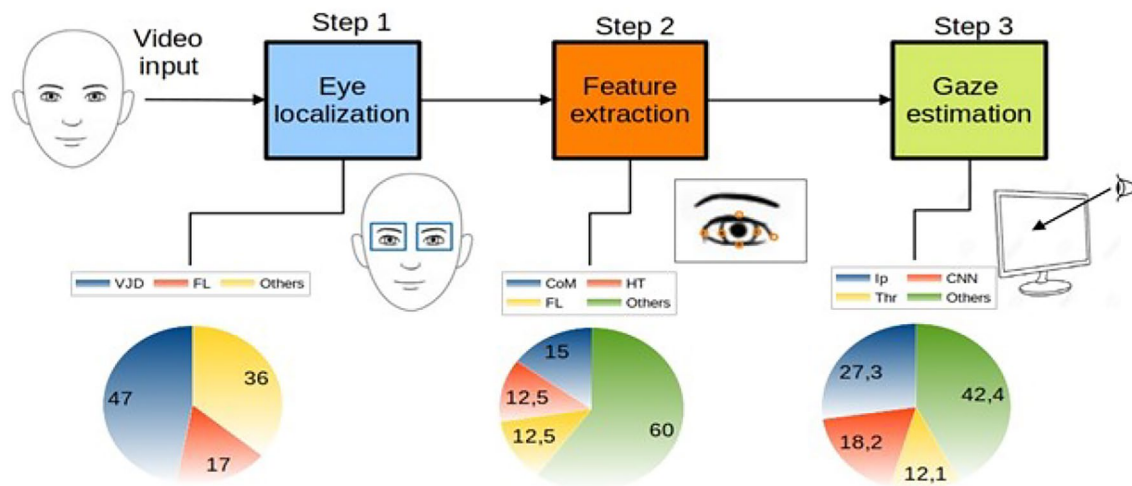
Method category	Step			Test	HP	II	Out.	Acc.	Ref	Comments
	1	2	3							
Feature-based (cont.)	VJD	HT,MS	Ip	2 Vid	-	×	↗	150–364 px	Santos et al. (2014)	Uses KF to stabilize pupil center KF is used to get more stable results. RANSAC for removing outliers. For head pose correction (Dementhon and Davis 1995) Eye centers and corners. Scale independent. Accuracy of 95% in predefined areas. BioID (BioID 2010). For illuminance invariant (Valenti and Gevers 2008; Lichtenauer et al. 2005) Low accuracy in vertical direction. FAST (Rosten and Drummond 2005) 5 eye features. Accuracy for error about 4 pixels. They prepared a DS: WEActivity Uses landmarks in eyes, eyelids, nose as features for classifying 5 eye gaze directions Several head postures have been also tested. Accuracy is related to 9 points on screen The accuracy was averaged among trials. Based on BioID dataset. Grad method: Timm and Barth (2011) Accuracy for normalized error <0.05 using Milborrow et al. (2010), RS (Loy and Zelinsky 2003). Datasets (FERET 2011; Milborrow et al. 2010; GTAV 2008; La Cascia et al. 2000) Uses the intercantal distance to normalize images. Different HP were tested and face aligned by SDM (Xiong and De la Torre 2013). Dataset (Smith et al. 2013a)
	FL, ASM, LK	CoM	LHM	5 Sub	✓	-	↗	-	Anwar et al. (2018)	
-	-	IsoC, GR	Ip	DS, 20 Sub	×	✓	↗	1.2°	Valenti et al. (2009)	
VJD	FAST IPROJ, Skw	Ip	10 Sub	×	-	↗	4.8°	Dung et al. (2018)		
VJD	CNN	-	DS	✓	✓	-	98.3–99.4%	Meng and Zhao (2017)		
FL, AAM	FL	SVM	15 Sub	×	×	↔	71%–94%	Wu et al. (2014)		
AdaB., BoI, GR	FFT2	SVM	12 Sub	×	✓	↗	91%	Lin et al. (2013)		
VJD Grad FAST	-	Thr	6 Sub, DS	-	✓	↔	91.5%	Wisniewska et al. (2014)		
VJD	ICLM RS	-	DS	≈	✓	-	92.9%	Skodras and Fakotakis (2015)		
HoG	FL	SVR CNN	19 Sub, DS	≈	-	↗	1.8°–4.6°	Tóser et al. (2016)		

Table 2 (continued)

Method category	Step			Test	HP	II	Out.	Acc.	Ref	Comments
	1	2	3							
VJD	ICLM, RS, TM, IPROJ, LK	Ip	Ip	12 Sub DS	≈	✓	↗	2.97 <sup>o</sup> *	Skodras et al. (2015)	It uses a patch to fix an anchor point, iris center (Skodras and Fakotakis 2015) and eyelid edge detection. Datasets: Weidenbacher et al. (2007); Smith et al. (2013a)
VJD, Dif	HT, TM, LK		Ip	1 Sub	✓	-	↗	83.3%	Khilari (2010)	Blink duration for left, right, or double click generation. BTE for head movement compensation. Dataset: Weidenbacher et al. (2007)
VJD	FL, CoM		-	7 Sub 11 PWD	×	×	⊠	-	Agarwal et al. (2019) and Sahay and Biswas (2017)	Authors measure the time to select 1 out of 4 items in a 2 × 2 grid. Results are compared to Webgazer.js
VJD	HT, RSWP		RFA	10 Sub	-	-	⊠	95.2%	Al-Btroush et al. (2019)	It slightly improves accuracy when using only eye centers
VJD	Haar, Thr		-	5 Sub	-	-	✦	86%	Chew and Penver (2019)	It compares eye-gaze directions with an internal database
FL	FL		-	10 Sub	✓	-	⊠	28.3%	Hossain et al. (2019)	It uses OpenFace (Baltrušaitis et al. 2016) to obtain FL and the head pose which are used as features for a artificial neural network (ANN)
FL	FL		LHM	6 Sub	×	×	↗	4.2°	Solska and Kocejko (2022)	Nine calibration point. Accuracy dependent on lighting

**Table 3** Continued from Tables 1 and 2

Method category	Step			Test	HP	II	Out.	Acc.	Ref	Comments
	1	2	3							
Model-based	VJD	NFLG	LPro	DS	×	✓	↗	0.8°	Hammal et al. (2005)	Includes blinking detection. Datasets: FERET (2011), Yale (1997) and Cohn (2001)
	-	TM, EFitt	3D model	3 Vid	×	-	↗	< 1.8°	Jankó and Hajder (2012)	Gaze estimation by minimizing the quadratic error using L-BFGS method
	FL,CLM	ICF	3D model	23 Sub DS BioID	✓	-	☒	>98%	Kim et al. (2016)	Facial points are used for estimating head pose. 2 × 3 areas on screen. Accuracy in iris center 86.9% for error < 0.05. CLM (Asthana et al. 2014). HP (Heo and Savvides 2011)
	VJD, GEM	FL, GR	Ip, RANSAC	-	×	×	↗	2.3°	Jariwala et al. (2016)	Accuracy of 3.3° according to Technology (2015) for a 3 × 3 grid. FL (Kazemi and Sullivan 2014)
Others	-	IsoC, HoG	-	DS: BioID	-	-	-	89.5%	Vater and León (2016)	Cascade classifier similar to the one used in VJD. Accuracy for normalized error <0.05
FL	SNK, CF	-	DS	×	✓	-	-	94.4%	Xiao et al. (2018)	Metrics proposed in Jesorsky et al. (2001). Accuracy for normalized error <0.05 using BioID (2010) and other datasets (BioID 2010; TFV 2004; Villanueva et al. 2013)
VJD	HT	-	8 Vid	-	-	-	-	60–90%	Ray Sarkar et al. (2015)	Eye center detection by an improved HT
CNN	FL	Ip	12 Sub UnitEyes	×	×	↗	1.93°	Cheng et al. (2022)	-	It combines appearance model to find features, using CNN to detect eye landmarks. UnityEyes (Wood et al. 2016)



**Fig. 3** Illustration of the processing pipeline for feature-based ETs. The most used techniques in each step are also shown on a pie chart

the PDM model. In subspace subspace constrained mean-shifts (SCMS) (Mansanet et al. 2013; Saragih et al. 2009) the distribution of likely landmark locations was substituted with simpler parametric forms, reducing sensitivity to local minima and the detection of false outlying points.

The use of deep learning in detecting some eye landmarks was accomplished by Cheng et al. (2022). Authors obtained the landmarks associated to the iris and the eyelid through an hourglass network trained by a set of eye images obtained with UnityEyes (Wood et al. 2016). In that automatically generated dataset iris and eyelid coordinates are known.

The use of the histogram of oriented gradient (HOG) as feature descriptor is also common in image processing for simplifying the image information and, in this context, for delimiting the head position (Tóser et al. 2016). The task involves calculating the magnitude and direction of the gradient operator within  $8 \times 8$  patches, followed by the creation of a histogram containing 9 bins, typically corresponding to angles 0, 20, 40, 60, 80, 100, 120, 140, and 160 degrees. The histogram is then normalized by using  $16 \times 16$  sequential blocks (to reduce illumination influence) and eventually all the histograms are stitched together.

As alternative to VJD and FL some authors have combined the use of template matching (TM) with antropometric rules (AR) (Magee et al. 2008) or geometric eye model (GEM) (Jariwala et al. 2015, 2016) to locate the eye in the image. Others have taken benefit from natural blinking to find the region of interest (ROI) by differentiating consecutive frames and then thresholding for the highest values (Magee et al. 2008).

Applying the eye detection algorithms frame by frame may need important computational resources. To reduce it, some authors proposed the use of well-known tracking

techniques such as Lucas–Kanade (LK) (Ferhat et al. 2015; Anwar et al. 2018).

- *Step 2: Feature extraction or model fitting* For appearance-based methods, this step mainly contains the algorithms that allow a reduction in data dimension through principal component analysis (PCA) (Liu et al. 2016), Kullback–Leibler divergence (KLD) (Yang et al. 2019) or gaussian process latent variable model (GP-LVM) (Wojke et al. 2016). Since these methods need to process a high amount of data, the use of small images will reduce the execution time. The effect of the image size on accuracy, by cropping the original image, has been studied in Liu et al. (2019). Namely, an image of  $60 \times 36$  can be horizontally and vertically cropped 10 times without a lost in accuracy.

Both feature- and model-based methods search for the iris/eye center position. Several techniques exploit the grayscale difference between the pupil/iris and the iris/sclera along with their circularity. For example, authors in Wiśniewska et al. (2014) and Jariwala et al. (2016) obtained the image gradient (Jariwala et al. 2015), which shows a high magnitude in the sclera-iris interface, and a direction, which, in negative, points to the eye's center. Namely, the iris center,  $\mathbf{p}$ , is the point in the image that maximizes the cost function built with the dot product between the normalized gradients and the vectors that join  $\mathbf{p}$  to the gradient positions.

Similarly, Hammal et al. (2005) looked for the lower semi-circle that maximizes the sum of the dot products between the normal vectors of the contour and the image gradient. The use of RS (Loy and Zelinsky 2003) allows defining the most voted point of the gradient image (Skodras et al. 2015; Skodras and Fakotakis 2015). Other authors calculated the curvature of the isophote (a contour with identical gray scale) (Valenti et al. 2009; Vater and León



2016). The curvature depends on the second derivative of the isophote, that, combined with the normal vector, defines the position of the center. Alternatively, the detection of the circular contour of the iris can be accomplished by using the HT (Alva et al. 2017; Santos et al. 2014; Khilari 2010; Ray Sarkar et al. 2015), or by finding the best ellipse fitting (Jankó and Hajder 2012) (Fig. 3).

The use of template matching is a well-known technique in image processing that can be also applied to find the iris position. The template can be made of a simple binary pattern for an initial rough detection (Jankó and Hajder 2012), or obtained during the calibration stage (Ferhat et al. 2015; Sewell and Komogortsev 2010). The snakuscule (SNK) (Snake + Minuscule) exploits the properties of the sclera-iris interface to find the point that maximizes the difference between the average gray level of an outer annulus with respect to an inner disk. The use of color components of the YCbCr space allows locating the eye building on the fact that the Cr takes low values in the eye compared to the areas covered by skin (Kourkoutis et al. 2007). In Skodras and Fakotakis (2015) and Skodras et al. (2015) authors exploit the fact that CbCr components are different in eye and surrounding area. Furthermore, the eye iris and pupil have lower luminance than the sclera. By combining all these components through a mathematical model, an efficient eye map can be built for detecting the iris center.

Most authors use grayscale images in the processing pipeline. The iris center position can be obtained by searching the center of mass (CoM) of the darkest pixels in the eye area (Rondio et al. 2012; Anwar et al. 2018). Bilateral filters (Zheng and Usagawa 2018), which preserve edge) can be applied to filter noisy points in low-quality images. Similar results can be obtained by eroding and then dilating the image (Ahmed et al. 2019). Eyelashes and shadows are sources of noise that can bias the pupil CoM. To deal with this, the image is firstly divided into six sections by applying a multi-level thresholding procedure (Karamchandani et al. 2015). To remove regions that do not belong to the iris, the darkest candidates are convolved with a binary mask that has a size similar to the iris. The center of mass of the resulting image is the pupil's center.

Although the iris or the pupil center position are used as a main feature in most studies, some authors have also included its relative position with respect to some reference or anchor points. Generally, both canthus (or eye corners) are selected as anchor points. In Torricelli et al. (2008), the outer corners are located by searching the most external points in a binary image obtained by differentiating consecutive frames with blinks. The use of templates of the eye corners (Alva et al. 2017; Ferhat et al. 2015), the curvature of isophote (Valenti et al. 2009), the Gabor filter (Lee et al. 2011), geometric rules (GR) (Valenti et al. 2009; Tóser et al. 2016; Rosten et al. 2008) are also feasible procedures to find

those corners. In Skodras et al. (2015) the points above and below the pupil's center, in the border with the eyelids, are also included. These points can be obtained by IPROJ

Machine learning is an emerging technology that can be also used for the location of eye features. In Meng and Zhao (2017), Authors applied a CNN to detect five eye features (the pupil center, external and internal eye corners, and upper- and lower-middle eyelid points). Up to 12 number of relevant points were obtained in Cheng et al. (2022) using an hourglass CNN. Others have used SVM, based on FM (Wu et al. 2014) or the bidimensional fast Fourier transform (FFT2) (Lin et al. 2013) as features for detecting a reduced set of eye gestures or positions on the screen.

• *Step 3: Gaze estimation* The final mapping between the eye gesture and the cursor position on the screen can be mainly implemented in three different ways:

- **Regression.** This is the most used and probably the easiest method. Based on the use of an initial calibration grid, the gaze coordinates are then obtained by linear interpolation of the features associated to each calibration point (Santos et al. 2014; Valenti et al. 2009; Dung et al. 2018; Skodras and Fakotakis 2015; Khilari 2010). Sometimes, this interpolation includes quadratic expressions, procedures to remove outliers, such as the RANSAC algorithm (Jariwala et al. 2016), and the assumption that the face position does not change significantly from the position it had during the calibration phase (Zheng and Usagawa 2018). Following a slightly different approach, in Liu et al. (2019, 2016) the authors applied local learning (LL) to find the optimal coefficients that map an appearance vector into the training set and, with them, the cursor position on the screen. Other authors used the CoEn (Yang et al. 2019), a measure of similarity between an eye image and the ones in the training set. From that measure, the optimal weight vector is estimated and used to determine the gaze direction. Another approach for regression is based on GPR (Wojke et al. 2016; Ferhat et al. 2015), which allows obtaining the probability distribution of possible interpolation functions. Calibration data contribute to get a posteriori probability function that, combined with the use of Gaussian kernels, allows finding an optimal fit.
- **Machine learning techniques.** Regression models are based on minimizing the sum of squared errors. As an alternative, SVR looks for finding an optimal weight vector that minimizes the absolute error with the following constraints: that the absolute error (a) must be less than a given margin; and (b) includes a slack variable that penalizes data outside of that margin (Mansanet et al. 2013). Other authors trained an artificial ANN (Sewell and Komogortsev 2010), or CNN (Ansari et al. 2021; Huang et al. 2021; Roy and Chanda 2022) for appearance-based

methods, and a CNN for feature-based (Tóser et al. 2016) with linear activation neurons at the last layer. The last layer allows selecting positions that do not belong to the calibration points. general regression neural network (GRNN) is based on a 3-layer neural network in which every training sample will represent a mean to a radial basis neuron (Torricelli et al. 2008). The output is a weighted average of radial neurons according to a kernel basis (Gaussian in most cases).

- Geometrical projections. Here, the eye gaze is obtained thanks to applying some geometrical model involving the eye position, the head and the screen. In Hammal et al. (2005) authors established a relationship between the position of the iris center in the image and its projection on the computer screen. Other authors have used the vector between the pupil and the eyeball's center positions on the camera's plane (Anwar et al. 2018); or applied a rotation matrix, modeling the head pose, to determine the new region of interest, according to the calibration data (Kim et al. 2016), or by estimating the focal 3D position through the minimization of an energy function. This includes the calibration points on the screen and the eye 3D pupil center coordinates and anthropomorphic rules (Jankó and Hajder 2012).

There are studies aimed at detecting a small subset of eye gestures in order to determine the area of the screen the user is looking at, or the cursor control as it was a joystick. The use of thresholds (Alva et al. 2017; Ahmed et al. 2019; Karamchandani et al. 2015; Wiśniewska et al. 2014), or algorithms based on machine learning (Lee et al. 2011; Wu et al. 2014; Lin et al. 2013) are common in this context.

- *Test*: Eye tracking systems are appealing for multiple purposes, apart from enhancing communication capabilities. Consequently, a very low number of studies have considered the ET as an assistive solution and therefore recruited people with disabilities (PWD) for testing. In this revision, two authors proposed detecting left-right eye movements either to select elements distributed horizontally on a computer screen (Alva et al. 2017), or to input text in a scan-based program (Magee et al. 2008). In another study, PWD were asked to select the correct answer of an online quiz application by glancing at one out of four different positions on the screen (Agarwal et al. 2019). As can be seen, PWD tested a reduced version of what an ET really is.

For real-time tracking tests, the number of subjects enlisted in the experiments was variable (ranged between 3 Liu et al. 2016 and 33 Ahmed et al. 2019). Nevertheless, most studies used datasets or prerecorded videos for an offline analysis, which allows the comparison with other scientific works. Particularly, we found databases collecting different head poses, facial expressions (Cohn 2001) and illumination conditions: BioID (2010), FERET (2011)

and Yale (1997); and that may include the coordinates of a high number of representative landmarks, such as in MUCT (Milborrow et al. 2010) and talking face video (TFV) (TFV 2004). Creating a dataset is a burdensome task, since it requires to make manual annotations of the eyelid corners, pupil centers and other relevant eye feature's positions. To overcome this issue, UnityEyes (Wood et al. 2016) was developed: a software that allows generating a complete dataset containing human eye images and feature point coordinates. There are also datasets available: CVC eye-tracking DB (CVC) (Ferhat et al. 2014), gaze interaction for everybody (GI4E) (Villanueva et al. 2013) and Columbia gaze DS (CAVE) (Smith et al. 2013b).

- *head pose (HP)*: To obtain a better gaze estimation, the head position and orientation should be taken into account. Usually, head pose estimators involve many assumptions in order to achieve a realistic modeling (i.e. the shape and size of the head, the rotation angles of the eye, etc.). However, the high computational cost of complex HP estimators may not be in line with the requirements of some systems, and additionally small mistakes in pose estimation might accumulate errors in the final gaze tracking output.

Some authors have not included specific strategies to compensate the head pose, but they have trained, or directly tested, their algorithms with natural head movements. For example, in Skodras and Fakotakis (2015), authors tested an iris center detection algorithm using datasets, like Milborrow et al. (2010) and GTAV (2008), that include subjects under various illumination and head poses. The same authors, in another paper (Skodras et al. 2015), completed the eye tracker by adding reference points. They obtained good results with the datasets (Smith et al. 2013a; Weidenbacher et al. 2007) even without having considered the head pose in the mapping function.

There are features that show certain level of head pose invariance. In Khilari (2010) slight head movements were compensated by using the iris-BTE feature. The BTE is unique on face, visible for any head pose and stable among different facial expressions. It is extracted on the basis of anthropometric measures after locating the eye region. Similarly, in Meng and Zhao (2017) the iris position was obtained relative to several reference points: eye corners and middle upper/lower eyelids. These relative movements entail a certain level of invariance to head movements.

Greater robustness for head pose estimation can be attained by employing the Perspective-n-Point (PnP) algorithm (Dementhon and Davis 1995; Anwar et al. 2018) such as the "POSIT" algorithm. This algorithm determines the translation vector and rotation matrix, enabling the computation of the object's feature points or facial landmarks (Xiao et al. 2018) through scaling and iterative procedures. In Kim et al. (2016) the head pose was also estimated by facial marks along with the Heo and Savvide's method (Heo

and Savvides 2011), that generates a geometric facial model containing four lengths related to eyes, nose and mouth. This model contributes to a better estimation of slant and tilt angles. By applying supervised descent method (SDM) (Tóser et al. 2016), in Xiong and De la Torre (2013) the gaze estimation was extended with 3D head pose data. Authors used 49 facial marks as inputs to SDM. They built a 3D mean face model, rotated it, and successively minimized the angular error. The position of the pupil center served as an optional feature, and the intercanthal distance (ICD) for normalization purposes.

Some datasets have been released for research purposes (Funes Mora et al. 2014; Zhang et al. 2015; Sugano et al. 2014), containing different head poses and then facilitating the comparison among proposals.

- **illumiance invariant (II):** Authors have adopted several strategies to minimize the influence of scene illumination on accuracy. One method is based on turning the RGB information into their YCbCr components and then giving more relevance to the color plane than to the luminance dimension. In Skodras et al. (2015) and Skodras and Fakotakis (2015) authors combined the color information with the radial symmetry transform (Loy and Zelinsky 2003) to derive an eye map that emphasizes the iris area. This procedure was tested with various datasets (La Cascia et al. 2000; GTAV 2008) and it reported very robust under several illumination conditions. Additionally, the usage of ratios in the chromatic components along with empirical thresholds (Kourkoutis et al. 2007) reduces the dependence on lighting conditions, although not completely.

Under non-ideal lighting, one half of the face may be darker than the other half. Therefore, by parceling the face image and changing the detector parameter for each half, it is possible to obtain better results (Rezaei and Klette 2012). In Wiśniewska et al. (2014), authors applied a binarization procedure based on  $p$ -tile thresholding in which the  $p$  value was obtained by assessing the mode and mean of pixel intensities in both eye regions separately. In Torricelli et al. (2008) authors proposed a modified Hough Transform that includes a voting system for each circle. The most voted circumferences of each eye are then compared with the candidates of the other eye to choose the pair of circumferences with the biggest total vote. This process has a good behavior over different light conditions even with asymmetric illumination of the face.

For relative homogeneous eye images, derivative operators, such as Canny algorithm, are efficient for detecting the iris contour independently of lighting conditions (Torricelli et al. 2008). The use of use snakuscule (minuscule snakes) (Xiao et al. 2018) has the property of adapting to different illumination conditions. Similarly, Valenti et al. (2009) used isophotes for eye center detection. Previous studies have shown that this technique is robust to rotation and lighting

changes (Valenti and Gevers 2008; Lichtenauer et al. 2005). However, the shadows in the eye affect the isophote, resulting in unrealistic eye centers.

To reduce shadows and homogenize the face, the median gray-level value of the upper half of the face images can be assigned to the pixels with a gray level greater than the median value itself (Lin et al. 2013). The retinal filter (Doutsi et al. 2018; Hammal et al. 2005), in some extent, mimics the receptive fields of the human eye. Used in the pre-filtering stage, it reduces the variations of illumination and contour enhancement. The illumination invariance can be also achieved by training a deep neural network with a huge amount of images taken under different lighting conditions (Meng and Zhao 2017).

- **Output:** The reviewed literature reports two main types of outputs for the eye-gaze detection algorithms: point on screen ( $\nearrow$ ,  $\boxplus$ ) and joystick mode ( $\leftrightarrow$ ,  $\clubsuit$ ). The latter is the simplest form, which only allows the detection of a discrete set of gaze directions (Wiśniewska et al. 2014; Lee et al. 2011; Magee et al. 2008) that handle the mouse pointer as a joystick; while, in the former, the pointer can directly jump to any position on screen (Yang et al. 2019).

- **Accuracy:** For the assessment of the ET performances, most authors ask subjects to look several times at specific points spread out on the computer screen. The average error is then computed for all points and participants. Normally, this error is given in degrees or pixels. We have favored the use of the same unit (degrees) to ease the comparison. Thus, we have marked with an asterisk those cases where it was impossible to make the conversion. Other authors split the screen in equally-sized squares and give the results as a percentage in placing the eye gaze inside them.

Studies that are only focused on determining the eye centers, typically use the same public databases [BioID (BioID 2010), GI4E (Villanueva et al. 2013), etc.] as other studies. These databases include annotations with the position of the eye center and other relevant eye features. Accuracy is then given in terms of the percentage of the estimations lying inside a circle whose radius is normalized by the distance between the true eye centers (Jesorsky et al. 2001).

There are also datasets available for eye tracking: CVC (Ferhat et al. 2014), gaze interaction for everybody (GI4E) (Villanueva et al. 2013), Columbia gaze DS (CAVE) (Smith et al. 2013b).

### 3 Selection of studies

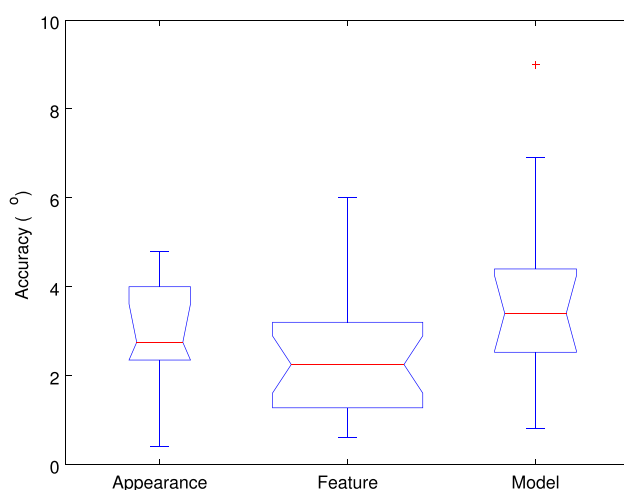
Previous section describes the state of art of the matter of this study, answering the (RQ1) in aspects concerning the most employed algorithms and the main features of the visible-light ETs. In order to find the criteria to continue with the study selection, several ET features must be taken

into account. The first, and maybe the most relevant one, is the accuracy. The better the accuracy, the better the ability of the ET to place the cursor at the correct position on the computer screen. There are factors influencing the accuracy negatively, so it would be also desirable that the algorithm shows certain robustness to head pose (to reduce the need of frequent calibrations) and to the scene lighting, that may cause many errors in detecting the eye centers.

Therefore, as the main key factor is the accuracy, is there then a method that stands out of the reminders? To answer this question (RQ2), we analyzed the significant differences among the feature-, appearance- or model-based methods. The accuracies in Tables 1, 2 and 3 along with the ones reported in Ferhat and Vilariño (2016), after removing duplicated entries, are shown in Fig. 4

The feature-based methods seem to have a slightly better figure in average ( $2.4^\circ$ ) with respect to the  $3.0^\circ$  and  $3.8^\circ$  of appearance- and model-based methods respectively. However, these differences were not statistically significant according to KW ( $p = 0.14$ ). This analysis included studies that allowed free head movements. Focused only on those studies that restricted head movements, the accuracy, in general, decreases, but even in this case, none method stands out from the rest significantly (KW,  $p = 0.21$ ). As a conclusion, there is no specific method that significantly improves the accuracy independently whether the head movements were constrained or not. Table IV collects the studies with best results in accuracy, grouped into three sets depending on whether the accuracy is less than, around or higher than  $3^\circ$ .

In a scenario in which the screen size is of  $18 \times 28$  cm (equivalent to  $13''$ , which is a normal size for many laptops and some tablets) and a subject-computer distance of 60 cm (in the range of most studies), the equivalent grid

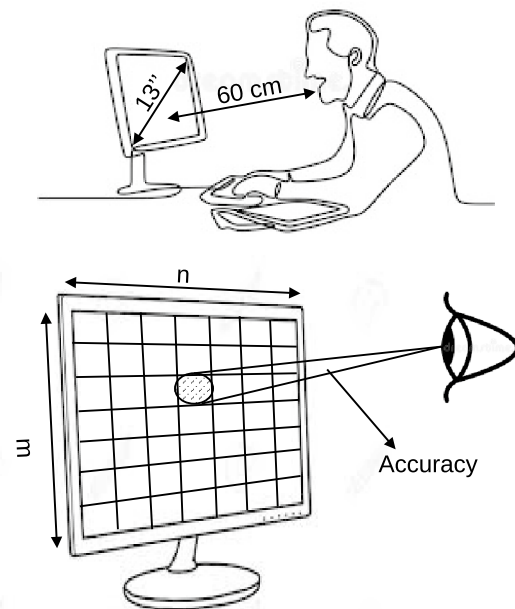


**Fig. 4** Accuracy for different visible light eyetracker techniques (Ferhat and Vilariño 2016). As can be seen, the feature-based technique gathers the highest number of published studies

resolution, ( $m \times n$ ) (Eq. 1) and cell side size (in cm) was estimated (Fig. 5). Both are dependent on the accuracy. Consequently, as the accuracy worsens, the grid cell size increases.

$$\begin{aligned} n &= \text{Round}\left(\frac{28}{d \times \text{tg}(Acc)}\right) \\ m &= \text{Round}\left(\frac{18}{d \times \text{tg}(Acc)}\right) \end{aligned} \quad (1)$$

Table 4 contains the studies with lowest accuracies (under  $2.3^\circ$ ), associated to a cell size less than 2 cm. None of them have considered the use of some technique to correct head



**Fig. 5** Illustration of the chosen computer screen size and user-screen distance for evaluating the grid dimension and cell size according to the accuracy

**Table 4** Summary of studies sorted by their accuracies

Acc.	Ref	HP	II	$m \times n$	Cell size (cm)
0.1	Yang et al. (2019)	$\times^{cr}$	$\times$	$172 \times 267$	$0.11 \times 0.11$
0.8	Hammal et al. (2005)	$\times$	$\checkmark$	$21 \times 33$	$0.84 \times 0.84$
0.95	Liu et al. (2016, 2019)	$\times^{cr}$	$\times$	$18 \times 28$	$1 \times 1$
1.2	Valenti et al. (2009)	$\times$	$\checkmark$	$14 \times 22$	$1.26 \times 1.26$
1.8	Jankó and Hajder (2012)	$\times$	$\times$	$10 \times 15$	$1.89 \times 1.89$
1.9	Wojke et al. (2016)	$\times$	$\times$	$9 \times 14$	$1.99 \times 1.99$
1.93	Cheng et al. (2022)	$\times$	$\times$	$9 \times 14$	$2.02 \times 2.02$
2.3	Jariwala et al. (2016)	$\times$	$\times$	$7 \times 12$	$2.41 \times 2.41$

The grid dimension and cell size for a user-computer distance of 60 cm and a screen size of  $13''$ . Five studies shown in bold were selected. The  $^{cr}$  symbol means that the authors used a chinrest

pose, but, at least, most of them have not constrained head movements in their experiments. Only Yang et al. (2019); Liu et al. (2019); Jariwala et al. (2016) used a chin rest and, for this reason, they were excluded. In Cheng et al. (2022) the distance between the screen and the participant was of 30 cm, but authors used a mobile phone instead of a computer, whose size is significantly smaller.

Studies with bigger cell sizes cannot operate like a true ET. These studies could be very useful for communication-oriented applications that show various symbols (icons) in a panel on the computer screen. By just looking at one cell in the panel, the user could directly select it instead of waiting for a scanning procedure to reach the desire icon.

## 4 Discussion

Research on eye tracking is increasing owing to its ability in facilitating many tasks in several areas such as: (1) Virtual and Augmented Reality Applications to enhance the user experience. In the future, it is expected a more widespread adoption of ET in VR/AR headsets, enabling more immersive and interactive experiences; (2) Healthcare and Medical Applications. Visible-light eye tracking has promising applications in healthcare to diagnose and monitor various conditions related to neurological disorders and eye-related diseases; (3) Market analysis and research, by providing valuable insights into user behavior, attention, and preferences, helping businesses and researchers optimize products and services; (4) Automotive Industry, where it is expected to play a significant role in the development of advanced driver assistance systems (ADAS), monitoring driver attention and fatigue, and therefore enhancing safety on the roads; (5) Accessibility. ET enhances accessibility for individuals with physical disabilities and create more intuitive user interfaces.

Many commercial ETs use IR-based elements, which have better performances than visible-light counterparts, but with a higher cost. A study verified this fact when both ETs were tested using several text entry applications (Chynał et al. 2010). Based on the data published in Kar and Corcoran (2017), the IR-ETs obtained an averaged accuracy of 1.8° while for the visible-light ETs it was higher, 3.1°, and such a difference was statistically significant -KW-test ( $p = 0.014$ ). The advantage of the visible-light ET is that only a camera is needed, and, since many laptops and tablets include at least one, there is no need for additional hardware.

The accuracy in visible-light ETs and IR -ETs is very sensitive to head movements. Reported data unveiled that, for IR-ETs, the accuracy worsened when head movements were allowed, 1.9° in comparison to when they were restricted or limited 1.6° (Kar and Corcoran 2017). In any ET An initial calibration phase is necessary to create a

map between the eye gaze with a position on computer screen. Therefore, any shift from that initial conditions can produce errors in the estimation of the eye gaze. Head movements during the operation phase or changing the relative position between the computer and the user, will worsen the initial mapping, and therefore influence the accuracy negatively. Consequently, frequent calibrations must be made in order not to reduce the performance of the ET. This repetitive procedure may become inconvenient for users due to the frequent disruption in the use of the tracking system. Several studies have proposed various solutions to reduce the burden of continuous calibrations. For instance, in Drewes et al. (2019), a quick calibration procedure based on smooth pursuit eye movements is implemented. The authors demonstrated that showing a moving point describing a circle for 4 s is sufficient to maintain good eye-tracking accuracy. Other methods operate recalibrating the eye gaze continuously by knowing the points on the screen selected after a mouse click or with the on-screen content (Huang et al. 2016; Gomez and Gellersen 2018). A common problem is the vertical drift when reading multiline passages. Several methods have systematically been reviewed in Carr et al. (2022) suggesting that dynamic time warping, which is a method for tackling drift and alignment problems, offers great promise. To alleviate the accuracy decrease during long-term use, the method proposed by Huang and Bulling (2019) analyzes the trajectories during saccades. It is based on the fact that saccade trajectories recorded by a calibrated ET are nearly straight between the saccadic start and endpoints. By observing multiple saccades between different on-screen locations, and by jointly minimizing saccade curvatures, calibration distortion can be reduced.

Any environmental condition that affects one of the steps needed for the implementation of the ET will also influence on accuracy. For example, the room illuminance and the shadows on the user's face reduce the algorithm's ability to locate the iris center and, therefore, the eye gaze accuracy. IR-based ETs usually have lighting elements placed in front of the face, so the image captured by the IR camera is commonly free of shadows (San Agustin et al. 2010; Pasarica et al. 2016; Kim et al. 2014). Conversely, the camera of IR-ETs does not work very well outdoors or whether the natural illumination is intense.

Machine learning is now proving its capability to enhance the performance of non-artificial intelligence solutions in numerous fields and applications (Zhang et al. 2023; Xie et al. 2023; Li et al. 2022). In this study, we have identified a machine learning approach (Wojke et al. 2016) that achieves an accuracy of under 2°, utilizing facial landmark (FL) to locate facial features and gaze point regression (GPR) for gaze estimation. A promising future lies ahead in this domain, with the development of more sophisticated

models that consider various head poses and lighting conditions. By doing so, eye tracking technology can become less sensitive to factors that might otherwise diminish accuracy.

Only three studies in this review included people with disabilities in their experimental tests. Two of them detected horizontal eye movements so that users can select the left or right area of the computer screen, as a binary input method in which only two choices are shown Magee et al. (2008) and Alva et al. (2017)). In Agarwal et al. (2019) the ET could detect a larger set of eye movements to select up to eight choices on a grid. Authors tested the ET using a reduced  $2 \times 2$  grid and compared it with a commercial IR-based ET and a free software called WebGazer.js. Results demonstrated that the proposed solution improves WebGazer but not the commercial IR-based device.

To access a computer, people with disabilities have several options. The simplest one is based on detecting a binary signal (on/off) coming from a mechanical button. Similarly, but more sophisticated systems are those that record bioelectrical signals or use video processing to generate those on/off signals. An example of such systems is those that detect voluntary blinks (Molina-Cantero et al. 2019), which are very useful when the motor functionality is quite affected. These binary detection systems need of specific applications that implement a scanning procedure for the selection of the elements contained on the computer screen. The scanning highlights every item on the screen for a dwell time. When the user presses the button, the highlighted item on the screen is selected. Depending on the application, the binary signal can be used to control a keyboard or a mouse (Molina-Cantero et al. 2021). Some interfaces emulate a joystick, allowing users to access the computer without needing for a scanning-type application (Roy and Chanda 2022; Rondio et al. 2012; Kourkoutis et al. 2007; Wu et al. 2014; Chew and Penver 2019). This method is faster than the scanning-based interfaces, owing to the item is directly accessed and it is not necessary to wait for the scanning to get to it, but slower than ETs. Molina-Cantero et al. (2021) compares several cursor control methods analyzing the time invested to select various targets, with different sizes, placed at different distances away from an initial position. The ET obtained the best result (Vertegaal 2008) compared to any brain computer interface (BCI) system, a joystick or even a mouse.

Once the cursor is placed at the desired position on the computer, the element addressed can be selected. To do that, some authors have proposed to keep the cursor still for a period (Sahay and Biswas 2017). However, this method prevents users from being focused on any element for long. Otherwise, the involuntary selection of the element will be performed. This is the so-called Midas touch effect (Velichkovsky et al. 2014). To avoid it, some ETs software has also included alternative selection methods such as, for example,

that based on the detection of voluntary blinks (Biswas and Langdon 2013).

A final aspect that has to be addressed in future, when implementing the selected ET for testing, is the execution time. In this study we have only found a few numbers of studies explained the computational load of their algorithms and in different ways that makes it difficult their comparison. Usually, the computer features, the operative system and the CPU percentage usage are described by many authors. Others give the processing speed of the system expressed in frames/s (Zheng and Usagawa 2018), which may be more appropriate to show the real-time capabilities of the proposed algorithm and favor the comparison among studies.

## 5 Conclusion and future work

This study contains a detailed review of the current state of visible-light eye tracking technologies, describing their main features and most used algorithms (RQ1). Three main methods (appearance-, feature- or model-based) were identified but none of them was statistically better than the others (RQ2). Head movements affect the accuracy negatively, and even though people with ALS show low amplitude of the head movements in advanced stages of the disease, they are enough to worsen the functionality of the ET. For this reason, only studies that did not use a chinrest were selected. The recommendations also include ETs with accuracies below  $2^\circ$ , for a user-computer distance of 60 cm and a screen size of 13 (RQ3). Higher accuracies mean higher on-screen grid sizes, and lower capacity in pointing small and near elements on it, such as icons, which reduces the functionality of the ET. Five studies fulfilled these recommendations, and only one was illuminance invariant.

The selected algorithms must now be tested at laboratory and real settings. The experiments will include the accuracy in detecting the pupil's centers with the use of several databases, the accuracy in positioning the cursor on the computer screen with several illumination conditions, and head poses. The final implementation could contain a mix of several parts of the selected algorithms, depending on the results obtained during each verification stage. Eventually, a usability test of the proposed solution will be accomplished by people with ALS.

**Acknowledgements** In memory of Moraima, a woman with ALS, whom we were honored to meet. We will be always grateful to you for your enthusiasm, willingness and patience. The authors would also like to thank Raquel Galán from ELA Andalucía (an association that attend people with ALS) for her support and collaboration in this work.

**Funding** Funding for open access publishing: Universidad de Sevilla/CBUA.

**Data availability** There is no data available for this research.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Agarwal A et al (2019) Comparing two webcam-based eye gaze trackers for users with severe speech and motor impairment. In: Chakrabarti A (eds) *Research into design for a connected world. Smart innovation, systems and technologies*, vol 135. Springer, Singapore. [https://doi.org/10.1007/978-981-13-5977-4\\_54](https://doi.org/10.1007/978-981-13-5977-4_54)
- Ahmed F, Rayhan MSS, Rahman S et al (2019) Controlling multimedia player with eye gaze using webcam. In: 2019 International conference on robotics, electrical and signal processing techniques (ICREST), pp 152–156
- Al-Btoush AI, Abbadi MA, Hassanat AB et al (2019) New features for eye-tracking systems: preliminary results. In: 2019 10th international conference on information and communication systems (ICICS), pp 179–184
- Alva M, Castellino N, Deshpande R et al (2017) An image based eye controlled assistive system for paralytic patients. In: 2017 2nd international conference on communication systems. Computing and IT applications (CSCITA). IEEE, pp 178–183
- Ansari MF, Kasprowski P, Obetkal M (2021) Gaze tracking using an unmodified web camera and convolutional neural network. *Appl Sci* 11(19):9068
- Anwar S, Milanova M, Abdulla S et al (2018) Emotion recognition and eye gaze estimation system: Erege. In: *International conference on human-computer interaction*. Springer, pp 364–371
- Ashby D (1991) *Practical statistics for medical research*. Douglas G. Altman, Chapman and Hall London, 1991. no. of pages: 611. price:£ 32.00. *Stat Med* 10(10):1635–1636
- Asthana A, Zafeiriou S, Cheng S et al (2014) Incremental face alignment in the wild. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1859–1866
- Baltrušaitis T, Robinson P, Morency LP (2016) Openface: an open source facial behavior analysis toolkit. In: 2016 IEEE Winter conference on applications of computer vision (WACV). IEEE, pp 1–10
- BioID (2010) The bioid face database. <https://www.bioid.com/About/BioID-Face-Database>
- Biswas P, Langdon P (2013) A new interaction technique involving eye gaze tracker and scanning system. In: *Proceedings of the 2013 conference on eye tracking South Africa*. ACM, pp 67–70
- Burton L, Albert W, Flynn M (2014) A comparison of the performance of webcam vs. infrared eye tracking technology. In: *Proceedings of the human factors and ergonomics society annual meeting*. SAGE Publications, Sage CA, Los Angeles, CA, pp 1437–1441
- Carletta J (1996) Assessing agreement on classification tasks: the kappa statistic. *Comput Linguist* 22(2):249–254
- Carr JW, Pescuma VN, Furlan M et al (2022) Algorithms for the automated correction of vertical drift in eye-tracking data. *Behav Res Methods* 54(1):287–310. <https://doi.org/10.3758/s13428-021-01554-0>
- Cheng S, Ping Q, Wang J et al (2022) Easygaze: hybrid eye tracking approach for handheld mobile devices. *Virtual Real Intell Hardw* 4(2):173–188
- Chew MT, Penver K (2019) Low-cost eye gesture communication system for people with motor disabilities. In: 2019 IEEE international instrumentation and measurement technology conference (I2MTC), pp 1–5
- Chynał P, Sobiecki J (2010) Comparison and analysis of the eye pointing methods and applications. In: *International conference on computational collective intelligence*. Springer, pp 30–38
- Cohn JF (2001) Cohn–Kanade au-coded facial expression database. <https://www.ri.cmu.edu/project/cohn-kanade-au-coded-facial-expression-database/>
- Cootes TF, Taylor CJ (1992) Active shape models—‘smart snakes’. In: *BMVC92*. Springer, pp 266–275
- Cootes TF, Edwards GJ, Taylor CJ (2001) Active appearance models. *IEEE Trans Pattern Anal Mach Intell* 23(6):681–685
- Dalmajier E (2014) Is the low-cost eyetribe eye tracker any good for research? Technical report, PeerJ PrePrints. Available online at: <https://peerj.com/preprints/585/>
- Dementhon DF, Davis LS (1995) Model-based object pose in 25 lines of code. *Int J Comput Vis* 15(1–2):123–141
- Doutsi E, Fillatre L, Antonini M et al (2018) Retina-inspired filter. *IEEE Trans Image Process* 27(7):3484–3499
- Drewes H, Pfeuffer K, Alt F (2019) Time- and space-efficient eye tracker calibration. In: *Proceedings of the 11th ACM symposium on eye tracking research & applications*, pp 1–8
- Dung LR, Lee YC, Wu YY (2018) On gaze estimation using integral projection of eye images. In: *International conference on human-computer interaction*. Springer, pp 159–167
- FERET (2011) Facial recognition technology. <https://www.nist.gov/itl/products-and-services/color-feret-database>
- Ferhat O, Vilarino F (2016) Low cost eye tracking. *Comput Intell Neurosci* 2016:17
- Ferhat O, Vilarino F, Sanchez FJ (2014) A cheap portable eye-tracker solution for common setups. *J Eye Mov Res* 1–10. <http://mv.cvc.uab.es/projects/eye-tracker/cvceyetrackerdbi>
- Ferhat O, Llanza A, Vilarino F (2015) A feature-based gaze estimation algorithm for natural light scenarios. In: *Iberian conference on pattern recognition and image analysis*, pp 569–576. [https://doi.org/10.1007/978-3-319-19390-8\\_64](https://doi.org/10.1007/978-3-319-19390-8_64)
- Funes Mora K, Monay F, Odobez JM (2014) Eyediap: a database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In: *Proceedings of the symposium on eye tracking research and application*, pp 255–258. <https://doi.org/10.1145/2578153.2578190>
- Gomez AR, Gellersen H (2018) Smooth-i: smart re-calibration using smooth pursuit eye movements. In: *Eye tracking research and applications symposium (ETRA)*. <https://doi.org/10.1145/3204493.3204585>
- GTAV (2008) Face database. <https://gtav.upc.edu/en/research-areas/face-database>
- Guennouni S, Ahaitouf A, Mansouri A (2015) A comparative study of multiple object detection using haar-like feature selection and local binary patterns in several platforms. *Model Simul Eng* 2015:17
- Hammal Z, Massot C, Bedoya G et al (2005) Eyes segmentation applied to gaze direction and vigilance estimation. In: *International conference on pattern recognition and image analysis*. Springer, pp 236–246
- Heo J, Savvides M (2011) Generic 3d face pose estimation using facial shapes. In: 2011 international joint conference on biometrics (IJB). IEEE, pp 1–8

- Hossain MS, Ali AA, Amin MA (2019) Eye-gaze to screen location mapping for ui evaluation of webpages. Association for Computing Machinery, New York
- Huang MX, Bulling A (2019) Saccalib: reducing calibration distortion for stationary eye trackers using saccadic eye movements. In: Proceedings of the 11th ACM symposium on eye tracking research & applications, pp 1–10
- Huang MX, Kwok TC, Ngai G et al (2016) Building a personalized, auto-calibrating eye tracker from user interactions. In: Proceedings of the 2016 CHI conference on human factors in computing systems, pp 5169–5179
- Huang J, Zhang Z, Xie G et al (2021) Real-time precise human–computer interaction system based on gaze estimation and tracking. *Wirel Commun Mob Comput* 2021:1–10
- Jankó Z, Hajder L (2012) Improving human–computer interaction by gaze tracking. In: 2012 IEEE 3rd international conference on cognitive infocommunications (CogInfoCom). IEEE, pp 155–160
- Jariwala K, Nandi A, Dalal U (2015) A real time robust eye center localization using geometric eye model and edge gradients in unconstrained visual environment. *Int J Comput Appl* 128:22–27. <https://doi.org/10.5120/ijca2015906415>
- Jariwala K, Dalal U, Vincent A (2016) A robust eye gaze estimation using geometric eye features. In: 2016 third international conference on digital information processing, data mining, and wireless communications (DIPDMWC). IEEE, pp 142–147
- Jesorsky O, Kirchberg KJ, Frischholz RW (2001) Robust face detection using the Hausdorff distance. In: International conference on audio-and video-based biometric person authentication. Springer, pp 90–95
- Kar A, Corcoran P (2017) A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. *IEEE Access* 5:16,495–16,519
- Karamchandani H, Chau T, Hobbs D et al (2015) Development of a low-cost, portable, tablet-based eye tracking system for children with impairments. In: Proceedings of the international convention on rehabilitation engineering & assistive technology, Singapore Therapeutic, Assistive & Rehabilitative Technologies (START) Centre, p 5
- Kazemi V, Sullivan J (2014) One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1867–1874
- Khilari R (2010) Iris tracking and blink detection for human-computer interaction using a low resolution webcam. In: Proceedings of the seventh Indian conference on computer vision, graphics and image processing. ACM, pp 456–463
- Kim H, Cha J, Lee WD (2014) Eye detection for near infrared based gaze tracking system. In: 2014 international conference on information science applications (ICISA), pp 1–3. <https://doi.org/10.1109/ICISA.2014.6847398>
- Kim HI, Kim JB, Lee JE et al (2016) Gaze estimation using a webcam for region of interest detection. *SIViP* 10(5):895–902
- Kourkoutis LG, Panoulas KI, Hadjileontiadis LJ (2007) Automated iris and gaze detection using chrominance: application to human-computer interaction using a low resolution webcam. In: 19th IEEE international conference on tools with artificial intelligence (ICTAI 2007). IEEE, pp 536–539
- La Cascia M, Sclaroff S, Athitsos V (2000) Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models. *IEEE Trans Pattern Anal Mach Intell* 22(4):322–336. <https://doi.org/10.1109/34.845375>
- Lee KB, Kim DJ, Hong KS (2011) An implementation of svm-based gaze recognition system using advanced eye region detection. In: International conference on computational science and its applications. Springer, pp 58–69
- Li H, Zeng N, Wu P et al (2022) Cov-net: a computer-aided diagnosis method for recognizing covid-19 from chest x-ray images via machine vision. *Expert Syst Appl* 207(118):029
- Lichtenauer J, Hendriks E, Reinders M (2005) Isophote properties as features for object detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE, pp 649–654
- Lin YT, Lin RY, Lin YC et al (2013) Real-time eye-gaze estimation using gaze with a low-resolution webcam. *Multimed Tools Appl* 65(3):543–568
- Liu Y, Lee BS, Sluzek A et al (2016) Feasibility analysis of eye typing with a standard webcam. In: European conference on computer vision. Springer, pp 254–268
- Liu Y, Lee BS, Rajan D et al (2019) Camtype: assistive text entry using gaze with an off-the-shelf webcam. *Mach Vis Appl* 30:407–421
- Loy G, Zelinsky A (2003) Fast radial symmetry for detecting points of interest. *IEEE Trans Pattern Anal Mach Intell* 25(8):959–973
- Magee JJ, Betke M, Gips J et al (2008) A human-computer interface using symmetry between eyes to detect gaze direction. *IEEE Trans Syst Man Cybern Part A Syst Hum* 38(6):1248–1261
- Mansanet J, Albiol A, Paredes R et al (2013) Estimating point of regard with a consumer camera at a distance. In: Iberian conference on pattern recognition and image analysis. Springer, pp 881–888
- Meng C, Zhao X (2017) Webcam-based eye movement analysis using cnn. *IEEE Access* 5:19,581–19,587
- Milborrow S, Morkel J, Nicolls F (2010) The MUCT landmarked face database. Pattern Recognition Association of South Africa <http://www.milbo.org/muct>
- Modi N, Singh J (2022) Real-time camera-based eye gaze tracking using convolutional neural network: a case study on social media website. *Virtual Real* 26:1489–1506
- Molina-Cantero AJ, Lebrato-Vázquez C, Merino-Monge M et al (2019) Communication technologies based on voluntary blinks: assessment and design. *IEEE Access* 7:70,770–70,798
- Molina-Cantero AJ, Castro-García JA, Gómez-Bravo F et al (2021) Controlling a mouse pointer with a single-channel eeg sensor. *Sensors* 21(16):5481
- Parada FJ, Wyatt D, Yu C et al (2015) Experteyes: open-source, high-definition eyetracking. *Behav Res Methods* 47(1):73–84
- Pasarica A, Bozomitu RG, Cehan V et al (2016) Eye blinking detection to perform selection for an eye tracking system used in assistive technology. In: 2016 IEEE 22nd international symposium for design and technology in electronic packaging (SIITME), pp 213–216. <https://doi.org/10.1109/SIITME.2016.7777280>
- Ray Sarkar A, Sanyal PG, Majumder S (2015) A hybrid approach for eye-centre localization for estimation of eye-gazes using low-cost web cam. In: 2015 IEEE international conference on research in computational intelligence and communication networks (ICRCICN), pp 273–278. <https://doi.org/10.1109/ICRCICN.2015.7434249>
- Rezaei M, Klette R (2012) Adaptive Haar-like classifier for eye status detection under non-ideal lighting conditions. In: ACM international conference proceeding series, pp 521–526. <https://doi.org/10.1145/2425836.2425934>
- Rondio J, Strumillo P (2012) A preliminary study on passive gaze tracking system for hci. In: 2012 joint conference new trends in audio & video and signal processing: algorithms, architectures, arrangements and applications (NTAV/SPA). IEEE, pp 133–138
- Rosten E, Drummond T (2005) Fusing points and lines for high performance tracking. In: Tenth IEEE international conference on computer vision (ICCV'05) vol 1. IEEE, pp 1508–1515
- Rosten E, Porter R, Drummond T (2008) Faster and better: a machine learning approach to corner detection. *IEEE Trans Pattern Anal Mach Intell* 32(1):105–119



- Roy K, Chanda D (2022) A robust webcam-based eye gaze estimation system for human–computer interaction. In: 2022 international conference on innovations in science engineering and technology (ICISSET). IEEE, pp 146–151
- Sahay A, Biswas P (2017) Webcam based eye gaze tracking using a landmark detector. Association for Computing Machinery, New York
- San Agustin J, Skovsgaard H, Mollenbach E et al (2010) Evaluation of a low-cost open-source gaze tracker. In: Proceedings of the 2010 symposium on eye-tracking research & applications. ACM, pp 77–80
- Santos R, Santos N, Jorge PM et al (2014) Eye gaze as a human–computer interface. *Procedia Technol* 17:376–383
- Saragih JM, Lucey S, Cohn JF (2009) Face alignment through subspace constrained mean-shifts. In: 2009 IEEE 12th international conference on computer vision. IEEE, pp 1034–1041
- Sewell W, Komogortsev O (2010) Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. In: CHI'10 extended abstracts on human factors in computing systems. ACM, pp 3739–3744
- Sharma FP, Jobanputra SY (2014) Augmentative and assistive communication in patients of locked-in syndrome: a case report. In: Jensen W, Andersen OK, Akay M (eds) Replace. Repair, restore, relieve-bridging clinical and engineering solutions in neurorehabilitation. Springer, Berlin, pp 735–740
- Skodras E, Fakotakis N (2015) Precise localization of eye centers in low resolution color images. *Image Vis Comput* 36:51–60
- Skodras E, Kanas VG, Fakotakis N (2015) On visual gaze tracking based on a single low cost camera. *Signal Process Image Commun* 36:29–42
- Smith B, Yin Q, Feiner S et al (2013a) Gaze locking: passive eye contact detection for human object interaction. In: ACM symposium on user interface software and technology (UIST), pp 271–280
- Smith BA, Yin Q, Feiner SK et al (2013b) Gaze locking: passive eye contact detection for human-object interaction. In: Proceedings of the 26th annual ACM symposium on user interface software and technology, pp 271–280
- Solska K, Koceljko T (2022) Eye-tracking everywhere-software supporting disabled people in interaction with computers. In: 2022 15th international conference on human system interaction (HSI). IEEE, pp 1–5
- Sugano Y, Matsushita Y, Sato Y (2014) Learning-by-synthesis for appearance-based 3d gaze estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1821–1828
- Technology T (2015) Tobii accuracy and precision test method for remote eye trackers. <https://stemedhub.org/resources/3310>
- TFV (2004) Talking face video. [https://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/talking\\_face/talking\\_face.html](https://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/talking_face/talking_face.html)
- Timm F, Barth E (2011) Accurate eye centre localisation by means of gradients. *VISAPP*, pp 125–130
- Torricelli D, Conforto S, Schmid M et al (2008) A neural-based remote eye gaze tracker under natural head motion. *Comput Methods Programs Biomed* 92(1):66–78
- Tóser Z, Rill RA, Faragó K, et al (2016) Personalization of gaze direction estimation with deep learning. In: Joint German/Austrian conference on artificial intelligence (Künstliche Intelligenz). Springer, pp 200–207
- Tricco AC, Lillie E, Zarin W et al (2018) Prisma extension for scoping reviews (prisma-scr): checklist and explanation. *Ann Intern Med* 169(7):467–473
- Valenti R, Gevers T (2008) Accurate eye center location and tracking using isophote curvature. In: 2008 IEEE conference on computer vision and pattern recognition. IEEE, pp 1–8
- Valenti R, Staiano J, Sebe N et al (2009) Webcam-based visual gaze estimation. In: International conference on image analysis and processing. Springer, pp 662–671
- Vater S, León FP (2016) Combining isophote and cascade classifier information for precise pupil localization. In: 2016 IEEE international conference on image processing (ICIP). IEEE, pp 589–593
- Velichkovsky BB, Rumyantsev MA, Morozov MA (2014) New solution to the midas touch problem: identification of visual commands via extraction of focal fixations. *Procedia Comput Sci* 39:75–82
- Vertegaal R (2008) A fits law comparison of eye tracking and manual input in the selection of visual targets. In: Proceedings of the 10th international conference on multimodal interfaces. ACM, pp 241–248
- Villanueva A, Ponz V, Sesma-Sanchez L et al (2013) Hybrid method based on topography for robust detection of iris center and eye corners. *ACM Trans Multimed Comput Commun Appl* 9(4):25:1-25:20. <https://doi.org/10.1145/2501643.2501647>
- Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. IEEE, pp I–I
- Weidenbacher U, G. Layher PMS, Neumann H (2007) A comprehensive head pose and gaze database. <https://trello.com/c/1CIA9jMC/3-uulmhpg-dataset>
- Wiśniewska J, Rezaei M, Klette R (2014) Robust eye gaze estimation. In: International conference on computer vision and graphics. Springer, pp 636–644
- Wojke N, Hedrich J, Droege D et al (2016) Gaze-estimation for consumer-grade cameras using a gaussian process latent variable model. *Pattern Recognit Image Anal* 26(1):248–255
- Wood E, Baltrušaitis T, Morency LP et al (2016) Learning an appearance-based gaze estimator from one million synthesised images. In: Proceedings of the ninth biennial ACM symposium on eye tracking research & applications, pp 131–138
- Wu YL, Yeh CT, Hung WC et al (2014) Gaze direction estimation using support vector machine with active appearance model. *Multimed Tools Appl* 70(3):2037–2062
- Xiao F, Huang K, Qiu Y et al (2018) Accurate iris center localization method using landmark, snacuscule, circle fitting and binary connected component. *Multimed Tools Appl* 77:25333–25353
- Xie T, Wang Z, Li H et al (2023) Progressive attention integration-based multi-scale efficient network for medical imaging analysis with application to covid-19 diagnosis. *Comput Biol Med* 159(106):947
- Xiong X, De la Torre F (2013) Supervised descent method and its applications to face alignment. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 532–539
- Yale (1997) Yale face database. <http://vision.ucsd.edu/content/yale-face-database>
- Yang B, Zhang X, Li Z et al (2019) An accurate and robust gaze estimation method based on maximum correntropy criterion. *IEEE Access* 7:23291–23302
- Zhang H, Tang J, Wu P et al (2023) A novel attention-based enhancement framework for face mask detection in complicated scenarios. *Signal Process Image Commun* 116:116985
- Zhang X, Sugano Y, Fritz M et al (2015) Appearance-based gaze estimation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4511–4520
- Zheng C, Usagawa T (2018) A rapid webcam-based eye tracking method for human computer interaction. In: 2018 international conference on control. automation and information sciences (ICCAIS). IEEE, pp 133–136