Research Article Open Access

Dario Cazzato*, Fabio Dominio, Roberto Manduchi, and Silvia M. Castro

Real-time gaze estimation via pupil center tracking

https://doi.org/10.1515/pjbr-2018-0002
Received September 20, 2017; accepted January 16, 2018

Abstract: Automatic gaze estimation not based on commercial and expensive eye tracking hardware solutions can enable several applications in the fields of human-computer interaction (HCI) and human behavior analysis. It is therefore not surprising that several related techniques and methods have been investigated in recent years. However, very few camera-based systems proposed in the literature are both real-time and robust. In this work, we propose a real-time user-calibration-free gaze estimation system that does not need person-dependent calibration, can deal with illumination changes and head pose variations, and can work with a wide range of distances from the camera. Our solution is based on a 3-D appearance-based method that processes the images from a built-in laptop camera. Real-time performance is obtained by combining head pose information with geometrical eye features to train a machine learning algorithm. Our method has been validated on a data set of images of users in natural environments, and shows promising results. The possibility of a real-time implementation, combined with the good quality of gaze tracking, make this system suitable for various HCI applications.

Keywords: gaze estimation; regression tree; appearance-based method; pupil tracking

1 Introduction

It has long been recognized that human interaction encompasses multiple channels [1]. Eye gaze plays a special role, as it can express emotions, desires, feelings and intentions [2]. Gaze tracking is the process of determining the point-of-gaze in the physical space. Accurate eye gaze tracking normally requires expensive specialized hardware (such as the eye-tracking solutions produced by Tobii [3] or SR Research [4]) that relies on active sensing (most commonly, infrared illuminators)[5]. This reduces the appeal of these systems for consumer market applications [6]. Moreover, these solutions often require a manual calibration procedure for each new user.

More recently, inexpensive solutions that do not require active illumination have been proposed [7]. These systems rely on modern computer vision algorithms. Some can use the camera embedded in any computer screen, laptop, and even tablet computer, requiring no additional hardware.

In this work, we propose a new algorithm that can estimate eye gaze in real-time without constraining the motion of the user's head. Our system does not need person-dependent calibration, can deal with illumination changes, and works with a wide range of distances from the camera. It is based on an appearance-based method that tracks the user's 3-D head pose from images taken by a standard built-in camera. From the same images, the irises are detected, and their center locations are fed (together with other geometrical measurements) to a machine learning algorithm that estimates the gaze point on the screen. Iris detection and gaze point estimation are computed in real-time.

Our system has been validated on a data set of images of users in natural environments, and shows promising results. In addition, we present qualitative results with an online user interaction test using our system. This experiment shows that the system can provide useful real-time information about the user's focus of attention. Other potential applications for this technology include data analytics on visual exploration from multiple users watching a video, and the control of assistive devices.

This paper is organized as follows. Sec. 2 illustrates the eye gaze estimation problem and related work. Sec. 3 describes the proposed method to achieve gaze estimation. The experimental setup is explained in Sec. 4.1, while the

***Corresponding Author: Dario Cazzato:** Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, L-1359 Luxembourg, Luxembourg, E-mail: dario.cazzato@uni.lu
Fabio Dominio: Urbana Smart Solutions Pte. Ltd.
Roberto Manduchi: Computer Engineering Department, University of California, Santa Cruz
Silvia M. Castro: Universidad Nacional del Sur, Bahía Blanca, Argentina

data set used to train the system is presented in Sec. 4.2. Results are shown and discussed in Sec. 4.3. Sec. 5 has the conclusions.

2 Related work

Following the seminal work of Just and Carpenter [8], who studied the relation between eye fixation and cognitive tasks, the measurement of eye gaze direction has been used in a broad range of application areas over time. This includes human-computer interaction (HCI) [9, 10], visual behavior analysis [11, 12], visual search [13], soft-biometrics [14, 15], market analysis [16], cognitive process analysis [17], and interaction with children affected by autism spectrum disorders [18, 19]. Moreover, eye gaze tracking is fundamental for human-robot interaction (HRI)[20, 21], as it provides useful information about user engagement, turn taking schemes, or intention monitoring. Gaze has been considered even in an environment with both robot and gaze-interactive displays [22], or for the design of Attentive Robots [23]. For a review of gaze estimation applications in HRI/HCI, socially assistive robotics (SAR), and assistive technologies, the reader is referred to [24].

The availability of modern low-cost depth sensors, combined with advances in computer vision, has lead to new solutions for gaze estimation that are less invasive and cheaper than prior methods, which were based on active illuminators. Passive gaze estimation solutions can be divided into two main categories: *model-based* and *appearance-based*. Model-based methods rely on a 3-D model of the head and of the eyeball and use geometric reasoning [25–29]. Their main advantage is that they can naturally handle head pose movements, provided that these can be measured reliably. Unfortunately, precisely locating the eyeball in space is very challenging; indeed, the most successful algorithms used a 3-D camera for this purpose [30, 31]. In addition, in the process of building the mathematical model of the eye, these methods need to know the relative pose of cameras and screen, as well as the relationship between multiple cameras and the parameters of each camera. Consequently, a small amount of noise can strongly influence the final estimation [32]. When compared with appearance-based methods, their accuracy is generally lower. In addition, it is unclear whether shape-based approaches can robustly handle low image quality [33].

In contrast, appearance-based methods detect and track one's eye gaze directly from images, without the need for a full 3-D model of head and eyeball. Instead, these methods learn a mapping function from eye images to gaze directions. They can manage lighting conditions changes and, since they normally use the entire eye image as a high-dimensional input feature and map this feature to low-dimensional gaze position space, can potentially work with low-resolution images [34], at the cost of acquiring a large amount of user-specific training data. User-dependent calibration is often necessary. A main problem with these methods is that they are not robust to head pose movements [35, 36], unless this is explicitly taken into account. For example, the method of Schneider *et al.* [37] achieves person-independent and calibration-free gaze estimation, at the cost of assuming fixed head pose. The approach of Ferhat *et al.*[38] is based on an iris segmentation algorithm in order to track anchor points; histogram features are then used in a Gaussian process estimator. This system requires a person-dependent calibration and cannot deal with free head movements.

In this paper we only compare our work with other projects that integrate 3-D head pose information and that achieve real-time estimations of gaze tracks. Tab. 1 presents a summary of the most relevant related methods. For each entry, we provide a description of the category (model-based or appearance-based), the input type, computational details, lighting conditions (when available), details about user-dependent calibration, error (as reported), and discussion of application in a real HCI/HRI scenario. Note that we only report light conditions in the case of online testing. For tests on existing data sets, we refer the reader to the description given in the data set documentation. We want to emphasize that this table is meant solely to provide some context through a snapshot of competing approaches. Quantitative comparative evaluation is complicated by the different experimental setups and data sets considered for benchmarking.

3 Gaze estimation method

Our system analyzes video frames from a regular camera to detect the pose of the user's head and the location (in the image) of facial features using an open source software (IntraFace [48]). Subsequently, the center of each iris is found using a customized version of the Circular Hough Transform (CHT) [49]. A random forest regressor, trained on a labeled data set, is then used to map pose and pupil center information into a gaze point on the screen in real-time. A block diagram of the proposed solution is shown in Fig. 1.

Table 1: Summary of relevant state of the art methods: m-b stands for model-based, while a-b stands for appearance-based, n.a. for not available data.

Method	Category	Input type	Computational Details	Lighting Conditions	User dependent calibration	Error	HRI/HCI scenario
Koutras and Maragos [39]	a-b	camera	n.a.	n.a.	yes	under 7°	offline video processing
Yoshimura *et al.* [40]	a-b	camera	n.a.	n.a.	no	95.3% (monitor is divided in 12 areas)	digital signage
Guo *et al.* [32]	a-b	camera	n.a.	natural light and fluorescent lamp	yes	5 – 8°	-
Lu *et al.* [41]	a-b	camera	potentially real-time	good illumination conditions	yes	2.49°	-
Sugano *et al.* [42]	a-b	multiple cameras	n.a.	n.a.	no	6.5°	-
Xiong *et al.*[43]	a-b	stereo camera	n.a.	natural light source	yes	6.43°	-
Funes-Mora and Odobez [44]	a-b	depth sensor	10 fps (only for gaze, not the whole solution)	n.a.	no	5.7 – 7°	natural dyadic interaction
Lu *et al.* [45]	a-b	camera	potentially real-time	n.a.	yes	3°	-
Wood *et al.* [27]	m-b	camera	12 fps	n.a.	no	6.88°	gaze estimation on tablet
Holland *et al.* [46]	a-b	camera	0.65 fps	n.a.	yes	6.88°	gaze estimation on tablet
Cazzato *et al.* [29]	m-b	depth sensor	8.66 fps	uniform, left or right	no	2.48°	soft-biometric identification
Sun *et al.* [28]	m-b	depth sensor	12 fps	n.a.	yes	1.38 – 2.71°	chess game, eye keyboard
Chen and Ji [47]	m-b	camera + 2 IR LEDs	20 fps	robust	no	1.78° (after 80 frames)	-
Zhang *et al.* [33]	a-b	camera	n.a.	n.a.	no	6.3° (cross-data set)	-

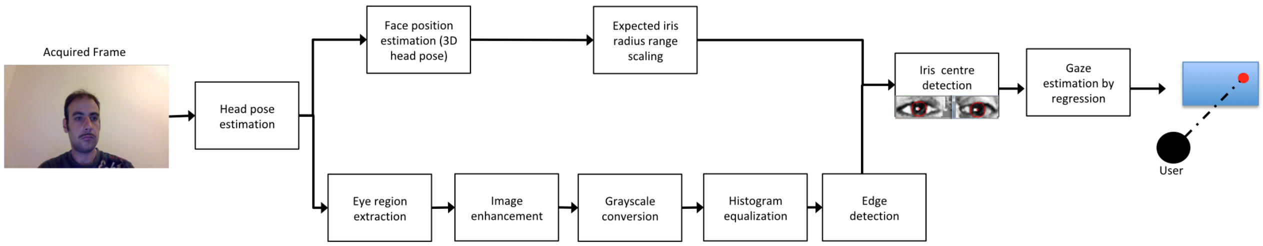


Figure 1: A block diagram of the proposed solution.

3.1 Head pose estimation

The head presence in the scene is detected by using the Viola-Jones face detector [50]. We use the IntraFace [48] software to detect face features from the image in real-time. The software also produces the head orientation (in terms of yaw, pitch and roll angles) with respect to a reference system centered in the camera. Head orientation is computed by aligning a deformable face model to the detected face. The model is characterized by the 2-D positions of a number of landmarks, describing the face, eye position (not pupils), mouth and nose outlines. Feature detection and tracking is based on the Supervised Descent Method (SDM), an algorithm that optimizes a non-linear least square problem. For more details, see [51].

The next step in our algorithm is the estimation of the vector T from the origin of the camera reference frame to a reference frame centered at the user's face. More precisely, the *head pose reference system* has its origin centered at the nose base and x-, y-axes parallel to the mouth and nose, respectively (Fig. 3). The rotation matrix R between the head pose and the camera reference system is produced by IntraFace. In order to compute T, we assume a fixed distance $D = 90$ mm between the external corners of the eye contours and a fixed distance $H = 70$ mm from a point at the bottom of the nose (*nose base*) and the segment joining the two pupils (see Fig. 2). Note that these feature points are computed by IntraFace. These values of distances between the selected features are justified by analysis of several studies. For example, the study of [52] determined that the interpupillary distance for the majority of humans lies in the range 50-75mm. More precisely, the average interpupillary distance for men is of 64.0 mm with a standard deviation of 3.6 mm and 61.7 mm with a standard deviation of 3.4 for women (2012 Anthropometric Survey of US Army Personnel [53]). Considering that the average palpebral fissure width is of approximately 30 mm, the distance between the external eye contours can be expected to be approximately of 94 mm for men and 91 mm for women. The chosen value of 90 mm is close to these average values. As for H, since no related data in the literature was found, this quantity has been chosen empirically.

The 3-D coordinates of the left and right eye corners, expressed in the camera reference system, are:

$$\begin{aligned} O_l^C &= \left[x_l^C, y_l^C, z_l^C\right]^T = R_C^H O_l^H + T, \\ O_r^C &= \left[x_r^C, y_r^C, z_r^C\right]^T = R_C^H O_r^H + T. \end{aligned} \quad (1)$$

In the head pose reference system, these vectors are expressed as $O_l^H = \left[D/2, H, 0\right]^T$ and $O_r^H = \left[-D/2, H, 0\right]^T$.

We express the rotation matrix R by its row components, i.e.

$$R_C^H = \begin{bmatrix} \mathbf{r_1} \\ \mathbf{r_2} \\ \mathbf{r_3} \end{bmatrix}. \qquad (2)$$

Let:

$$\begin{aligned} z_l^C\,[u_l, v_l, 1]^T &= KO_l^C \\ z_r^C\,[u_r, v_r, 1]^T &= KO_r^C \end{aligned} \qquad (3)$$

be the projections on the camera's focal plane of the left and right eye corners, where K is the intrinsic camera matrix, and u, v are the image coordinates of the left (subscript l) and right (subscript r) pupil center locations.

Putting these equations together, Eq. 3 can be rewritten as:

$$\begin{cases} x_l^C = z_l^C \frac{u_l - c_x}{f_x} \\ y_l^C = z_l^C \frac{v_l - c_y}{f_y} \\ x_r^C = z_r^C \frac{u_r - c_x}{f_x} \\ y_r^C = z_r^C \frac{v_r - c_y}{f_y} \end{cases} \qquad (4)$$

Thus, it is possible to compute the translation vector T, considering both eyes, as:

$$T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{cases} z_l^C \frac{u_l - c_x}{f_x} - \mathbf{r_1} O_l^H \\ z_l^C \frac{v_l - c_y}{f_y} - \mathbf{r_2} O_l^H \\ z_l^C - \mathbf{r_3} O_l^H \end{cases} = \begin{cases} z_r^C \frac{u_r - c_x}{f_x} - \mathbf{r_1} O_r^H \\ z_r^C \frac{v_r - c_y}{f_y} - \mathbf{r_2} O_r^H \\ z_r^C - \mathbf{r_3} O_r^H \end{cases} \qquad (5)$$

This directly provides the component z_l^C as:

$$z_l^C = \frac{f_x \mathbf{r_1}(O_l^H - O_r^H) - \mathbf{r_3}(O_l^H - O_r^H)(u_l - c_x)}{u_l - u_r} + \mathbf{r_3}(O_l^H - O_r^H) \qquad (6)$$

from which the vector T is easily computed.

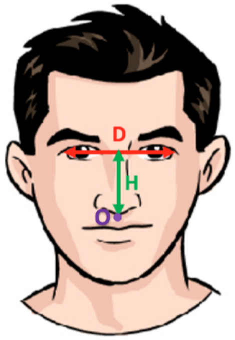


Figure 2: A scheme of the employed face measurements. Point O lies on the nose base.

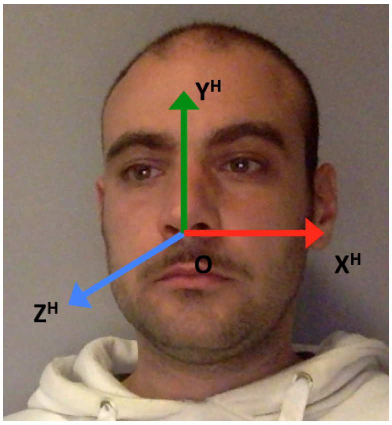


Figure 3: Head pose coordinate system axes position and orientation.

3.2 Iris detection

IntraFace produces several key points in the periocular region. Within these regions, we extract the iris areas, whose centers are then used for gaze estimation. We start from the bounding boxes of the two periocular regions, as provided by IntraFace, which has been augmented by 10 pixels in each dimension to compensate for alignment errors. The greyscale image is low-pass filtered to reduce noise; then, it is high-pass filtered and histogram equalized.

A Canny edge detector [54] is used to extract the irises edges. The parameters of the Canny edge detector need to be chosen carefully, to avoid the risk of returning a large number of false positives (the small blood vessels in the sclera) or, conversely, to miss part of or the entire iris contour. In our experiments, we used a Gaussian kernel of 15 × 15 pixels with $\sigma = 2$. Note that, due to the histogram equalization, the distribution of intensity values within each eye region covers the full available range, and thus adaptive thresholds are not required. The current implementation sets the minimum and maximum thresholds of the Canny edge detector to 60 and 128 respectively, which roughly correspond to 1/4 and 1/2 of the full range [0, 255].

The iris regions appear circular in the image when the user is imaged front-to-parallel and ellipsoidal otherwise. We use the Hough transform to extract circles at multiple radii from the edge image [49]. Specifically, given an $M \times N$ pixel area, for each radius R_{iris} of the circle under consideration, a counter is defined at each pixel. An edge pixel at $p = (p_x, p_y)$ triggers an increment by 1 of all counters at pixels $\{p + r\}$, where r spans the circumference of radius R_{iris} centered at the origin. The counter with highest value over all radii determines the estimated iris circle. In our implementation, we look for circumferences with radius R_{iris}

in the range $[r_A, r_B]$, with $r_A = 5$ mm and $r_B = 7mm$. We also experimented with a version of the Hough transform that extracts ellipses; however, due to the larger number of parameters, this version was too slow for real-time implementation, without appreciable benefits.

This iris detection algorithm can be improved by observing that the iris is typically darker than the surrounding sclera. This means that, at the iris' edge, the image gradient is expected to point outwards. We exploit this observation by only incrementing a counter at $p + r$ when it is *compatible* with the image gradient, that is, when the vector r forms an obtuse angle with the image gradient at p. This strategy has proven effective, but computing this angle introduces a substantial computational cost. This can be alleviated by quantizing the vector r into angular multiples of 45°, and maintaining, for each quantized angular value, a look-up table that determines which gradient angles are compatible with r. We also experimented with assigning different incremental values to the counters depending on the magnitude of the gradient at the edges, but this didn't result in an appreciable improvement.

We implemented one more variation to the original Hough algorithm, one that still exploits the property that the iris is typically darker than the surrounding sclera. This approach selects a Hough peak by taking into account not only the value of a counter, but also the average brightness within the candidate circle. Specifically, we keep 10% of the candidate circles with highest value of the associated counter; among these, we select the one with the lowest average brightness value within its boundary. Figs. 4 and 5 show some examples of iris detection with our algorithm.

Our iris detection algorithm is applied to both the left and right eye image region. Let (u, v) represent the pixel coordinate of the iris center (the pupil) for one of the two eyes. Our next step is to transform this location into a 3-D point expressed in the head reference system. Denoting the iris center position in camera coordinates by p^C and in head pose coordinates by p^H, the following relationship holds:

$$p^C = \alpha K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{7}$$

where K is the intrinsic calibration matrix and α is an unknown scale factor. Then,

$$p^H = R^H_C p^C + T^H = R^H_C p^C - R^H_C T^C, \tag{8}$$

where the notation $R^y_x = (R^x_y)^T$ indicates the rotation matrix from the coordinate system x to the coordinate system y, and T^x is a translation vector in the coordinate system x. p^H lies in the straight line s.t.

$$p^H = \alpha R^H_C p^C - R^H_C T^C. \tag{9}$$

This line intersects the plane $z^H = 0$ at (using Eqs. 7 and 8):

$$\alpha R^H_C p^C - R^H_C T^C = 0 \iff \alpha R^H_C \left(K^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix}^T \right) - R^H_C T^C = 0 \tag{10}$$

This implies, decomposing R^H_C in three vectors $\mathbf{r_1}$, $\mathbf{r_2}$ and $\mathbf{r_3}$:

$$\begin{gathered} \alpha \mathbf{r_3} \left(K^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix}^T \right) - \mathbf{r_3} T^C = 0 \Rightarrow \\ \Rightarrow \alpha = \frac{\mathbf{r_3} T^C}{\mathbf{r_3} \left(K^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix}^T \right)}. \end{gathered} \tag{11}$$

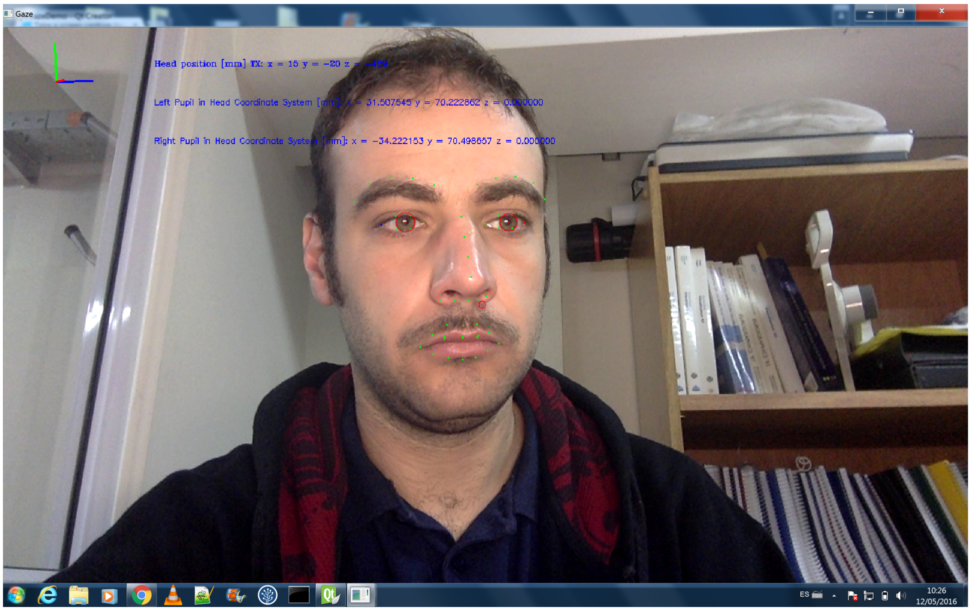
Substituting in Eq. 9, one obtains the pupil center location p^H in the head pose coordinate system.

3.3 Gaze estimation by random regression forests

The algorithm described above results in the location (in the image plane or in the head coordinate system) of the two iris centers. The next step is to map this information (together with the head pose) into a gaze point on the screen. We design a separable regressor to compute the (x, y) screen coordinates of the gaze point (one independent regressor per coordinate). We use a random forest regressor [55], based on a combination of decision tree predictors [56] such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [57]. When used as a regressor, the random forests algorithm works as follows:

1. Take n_{tree} bootstrap sample from the data by random sampling with replacement from the original training data, where n_{tree} represents the number of trees in the forest.
2. Grow, for each sample, an unpruned regression tree where at each node, m variables at random out of all M possible variables are selected independently at each node, and the best split is chosen on the selected m variables.
3. Predict new data by aggregating the predictions (average value) of the n_{tree} trees.

An estimation of the error can be obtained at each bootstrap iteration by using the tree grown with the bootstrap



```
Head position [mm] TX: x = 15 y = -20 z = -489

Left Pupil in Head Coordinate System [mm]: x = 31.507545 y = 70.222862 z = 0.000000

Right Pupil in Head Coordinate System [mm]: x = -34.222153 y = 70.498657 z = 0.000000
```

Figure 4: The proposed feature extraction algorithm for an incoming frame. Green points represent IntraFace facial tracking output, while red circles on the eyes represent iris detection output.

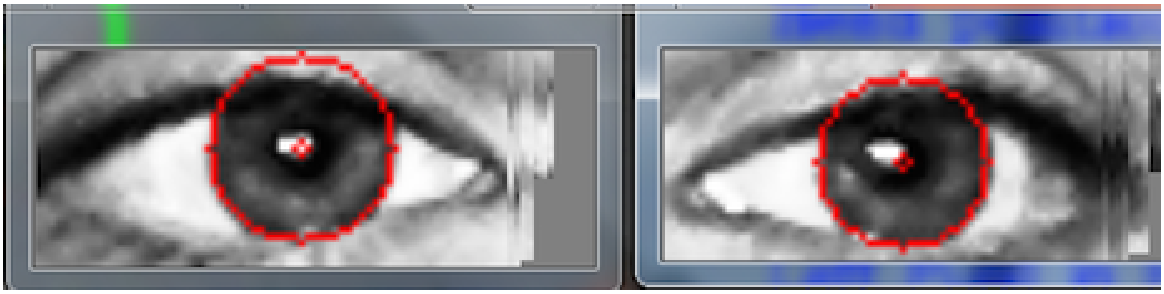


Figure 5: Sample results from our pupil center detection algorithm.

sample to predict the data that does not lie in the bootstrap sample (i.e. the *out-of-bag predictions*, see [58] for more details).

The random forest is trained on labeled data as explained in the next section.

4 Experiments

In this section we discuss our experiments. In particular, the experimental setup is discussed in Sec. 4.1, while Sec. 4.2 introduces two different data sets for our experiments. Results of the experiments are shown in Sec. 4.3.

4.1 Experimental setup

For our experiments, we used a laptop (a MacBook 15" with retina display, 2.6 GHz Intel Core i7 processor, and 16 GB 1600 MHz DDR3 of memory) with built-in camera. The laptop was placed on a desk during the experiments. Images were acquired at a resolution of 1280 × 780 at 30 fps, while the screen resolution for the test was set to 2880 × 1800. The system has been tested on a Windows 7 machine with 8 GB of RAM on a virtual machine run by Parallels Desktop v10.

The software was implemented in a single threaded C++ application using OpenCV [59] and Qt libraries [60] for the user interface. Facial features are detected and tracked by the IntraFace library [48], that also returns the head rotation in real-time. The random forest regressor used the OpenCV implementation with the following parameters:

maximum tree depth = 25; maximum number of tree in a forest = 200; size of the randomly selected subset of features at each node used to find the best split = 4; minimum number of samples required at a leaf node for it to be split = 5.

4.2 Data sets

We have used two different data sets in our experiments. The first set (Data Set 1) comprises six short videos from two different users. The users were asked to look at a circle moving in the screen in order to capture and evaluate different eye positions. These videos were 20 seconds long, and all frames have been manually labeled. The users' faces were illuminated by a desktop lamp positioned behind the camera. This data set was used to assess the error in the estimation of the irises' position in the image, as described in Sec. 4.3.1.

The second set (Data Set 2), which was used to evaluate the mapping between pupil center and gaze point, is composed by 1130 images taken from 10 different participants. 70% of the images in this data set were acquired with a light source coming from a desk lamp positioned behind the screen. The remaining remaining images were acquired with artificial room illumination, thus reproducing common user interaction scenarios in an indoor environment. Participants were asked to look at a colored marker (a red circle with radius of 90 pixels, corresponding to a field of view of 0.8°) appearing at random locations on a black screen. When a circle is displayed, participants were tasked with looking at it and pressing a key on the keyboard. Then the circle was reduced in diameter by two thirds, at which point participants pressed a second key. The system acquired and stored, along with the position of the circle center in the screen, the left and right irises' position in the camera and in the head pose reference systems.

Each participant was tested with both illumination conditions. Participants took turns in the data collection, with each participant testing with 10 circle locations before another participant took over. Note that the rotation matrix R_C^H and translation vector relating the screen and the camera reference systems are also stored in the file (the origin of the screen reference system is placed at the top left corner of the screen).

4.3 Experimental results

With the aim to provide a complete analysis of the solution, our system evaluation was divided into two different steps. In the first test, we assessed the iris detection system (Sec. 4.3.1). In the second one, we evaluated the performance of the gaze estimation algorithm by means of leave-one-out cross-validation (Sec. 4.3.2). A qualitative evaluation of the system in an HCI scenario is described in Sec. 4.3.3. Several possible sets of feature vectors have been analyzed and their performance has been compared.

4.3.1 Iris detection

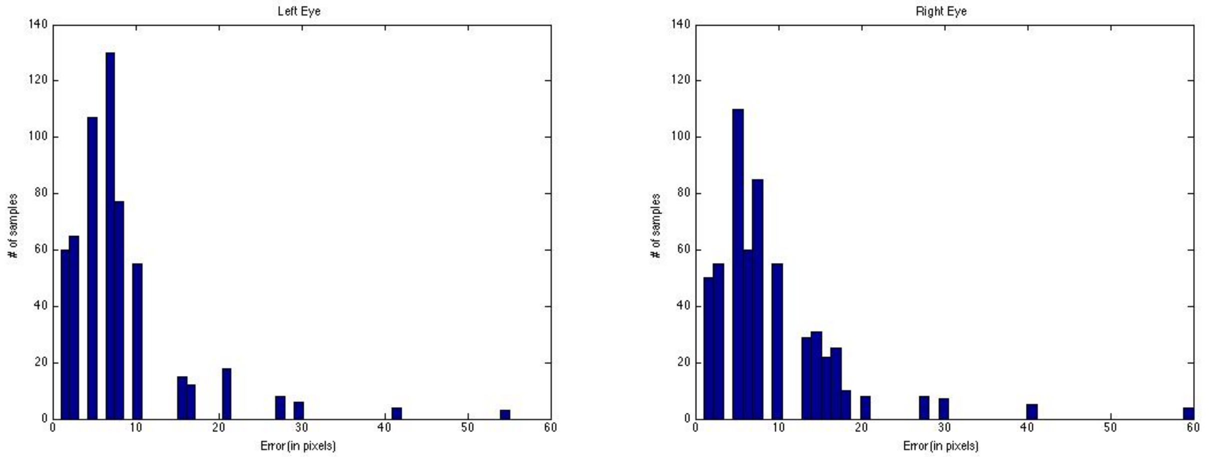
Data Set 1 (see Sec. 4.2) was used for this test. At each frame, we measured the distance E (in pixels) between the location of the center of the detected circle and the manually labeled pupil center for both the left and right eye. Fig. 6 shows the histograms of E over all frames and users for the left and right eye. Note that computing the center of both irises (as explained in Sec. 3.2) takes about 5.4 msec in our implementation.

It should be noted that the precision of iris center estimation is affected by the quality of the acquired images, the limited image size of the periocular regions, and the illumination changing conditions (including the shadows on the head surface that are generated by head movement).

With respect to other algorithms in the literature that may achieve higher localization accuracy [61–63], our system has the advantage that it can be implemented at a high fame rate, and thus represents an attractive solution for those situations in which speed is more critical than high precision.

4.3.2 Gaze point detection

We used Data Set 2 to assess the quality of the mapping from iris center measurement to gaze point using regression random forests. We used two cross-validation modalities. In the first modality, each vector in the data set was selected in turn as a test vector, and the system was trained on all remaining vectors. Error statistics are shown by means of histograms in Fig. 7. The feature vector given in input to the regressor includes the 3-D head pose (rotation and translation) as well as the irises' locations. We considered different representations for these quantities, specifically: Euler angles vs. quaternions for the head rotation, reference systems for the irises' location. In addi-



(a) Left eye errors

(b) Right eye errors

Figure 6: Iris detection performance histograms.

tion, we experimented both with single (left) pupil center and with both centers. Note that the feature length varied between 8 and 11, depending on the representation chosen and on whether one or two eyes were considered. The best results were found when using both pupils and the quaternion representation of the head rotation. Fig. 8 shows results in terms of gaze point errors in the *x* and *y* coordinate. Note that, in general, detection accuracy tends to be higher for the *x*-axis.

In the second cross-validation modality, we selected the data from each participant in turn as a test data set, and trained the system with data from the other participants. For these tests, we used the configuration with the head rotation expressed using quaternions, and pupils centers position expressed in terms of the head pose reference system. Tab. 2 shows the results for each participant.

Tab. 3 compares the average mean square error using our system (evaluated with the first cross-validation modality) against similar results for other real-time systems reported in the literature. Results are expressed in units of angular error for consistency.

Our experiments have highlighted a noticeable difference in accuracy between the *x* component (mean error = 2.29°) and the *y* component (mean error = 5.33°). The main reason for this behavior is the higher error in the *y* component of iris localization.

4.3.3 Qualitative results

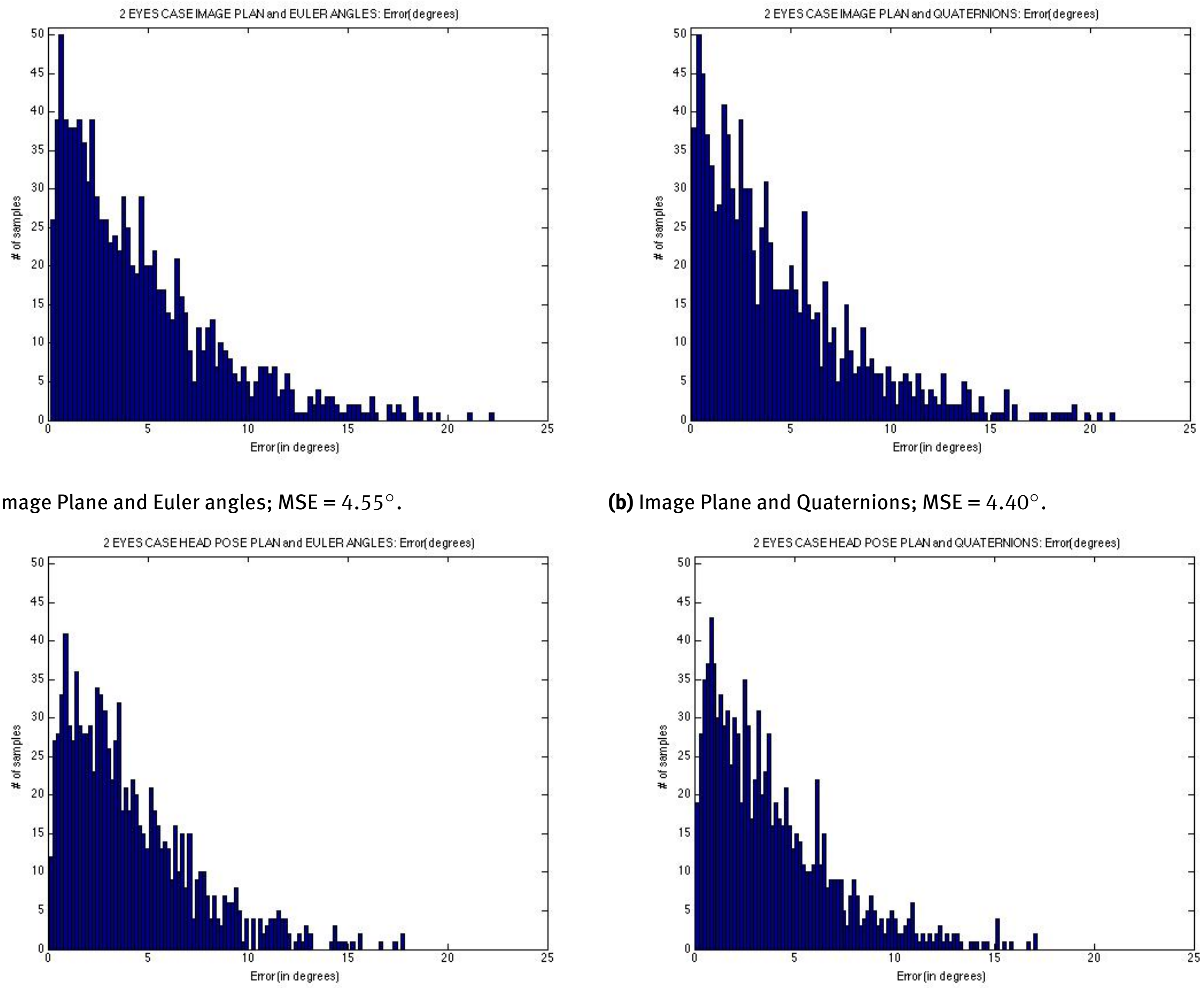
We conducted a qualitative test in order to assess how our system could be used for a real-world human-computer interface (HCI) application. In this simple test, two videos with a target (represented by the red circle) moving on the screen was shown to three participants. In the first video, the circle was moving on a simple rectangular trajectory, while in the second video the circle followed a sawtooth trajectory. Participants sat in front of the screen at a distance varying between 40 and 70 cm. Uniform illumination was created by artificial light in the room. Participants were asked to follow the circle's trajectory with their gaze. They were informed prior to the test about the details of the test, and about the expected trajectory of the circle.

Fig. 9 shows the circle's trajectories (blue line) and the measured gaze point trajectories (red line) for user #1. Fig. 10 shows the complete hit maps for all three users interacting with our system with both videos. Each user is represented by a different color, with the brightness of the color (light to dark) indicating the progress of the trajectory.

Note that our system, in the configuration considered for this test, can process images at 8.88 frames per second on average, for an input resolution of 1280 × 780 pixels. This frame rate makes it suitable for various real-time HCI applications [24, 29, 64–66]. When a faster tracking rate is required, hardware-based gaze tracking solutions should be used [67–70].

5 Conclusion

We proposed a novel real-time gaze estimation system suitable for HCI applications. This system uses a regular camera, of the type that is typically embedded in laptops and



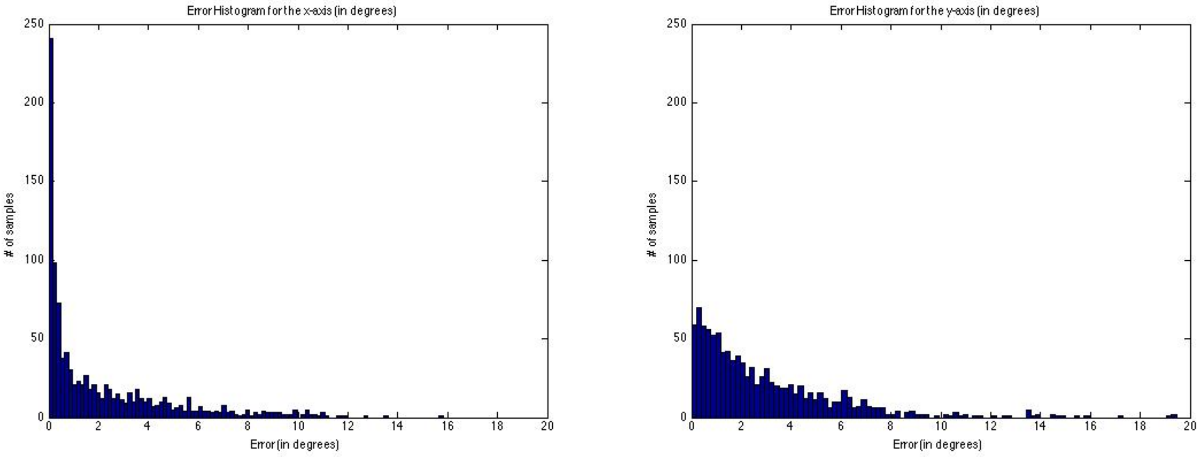
(a) Image Plane and Euler angles; MSE = 4.55°.

(b) Image Plane and Quaternions; MSE = 4.40°.

(c) Head Pose Plane and Euler angles; MSE = 4.04°

(d) Head Pose Plane and Quaternions; MSE = 3.81°.

Figure 7: Data Set #2 validation results for different short feature vector combinations with 2 eyes: errors are expressed in degrees.



(a) Error on the x-axis (in degrees).

(b) Error on the y-axis (in degrees).

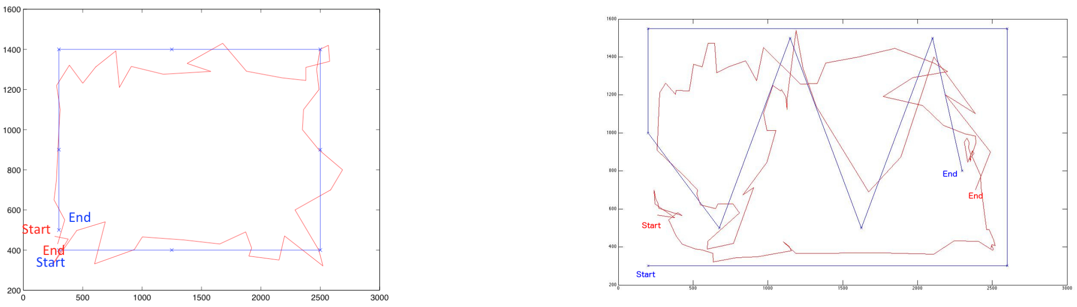
Figure 8: Error analysis by using two eyes and expressing rotations by quaternions.

Table 2: Mean square eye gaze direction error (in degrees) with a truncated Data Set 2 with a person excluded from the training set.

Person #	Error (degrees)
1	3.99°
2	4.01°
3	3.65°
4	3.68°
5	3.90°
6	4.02°
7	4.12°
8	3.86°
9	3.75°
10	3.91°

Table 3: Mean square eye gaze direction error (in degrees) compared with other published real-time systems.

Method	FPS	Reported Error	Test Distance (≈)	Additional Requirements
Proposed Method	8.88	3.81 °	40-70 cm	–
Wood and Bulling [27]	12	6.88°	20 cm	20 cm distance only
Holland *et al.* [46]	0.65	3.95°	50 cm	5 mins of personal user calibr.
Cazzato *et al.* [29]	8.66	2.48 °	70 cm	Depth sensor
Sun *et al.* [28]	12	1.38 – 2.71°	55 cm	Depth sensor & personal user calibr.
Chen and Ji [47]	20	1.78°	45-70 cm	2 IR LEDs (error for first 80 frames is 2.01°)



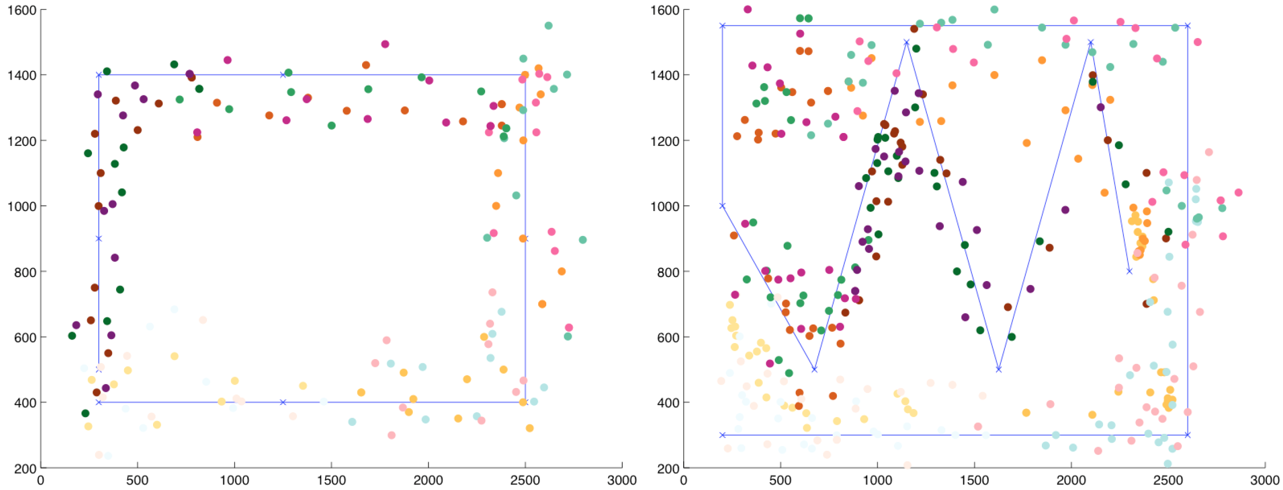
(a) Generated (blue) and estimated (red) gaze points (rectangular pattern).

(b) Generated (blue) and estimated (red) gaze points (complex trajectory).

Figure 9: Example of real-time interaction with user #1 and two predefined sequences.

computer screens. The proposed system does not require a user-dependent calibration, can deal with illumination changes, and can work with a variety of head poses. The solution is based on an appearance-based method that uses video from a regular camera to detect the pose of the user's head and the location (in the image) of the eye features. This information is fed to a machine learning system, which produces the gaze point location on the screen. Our end-to-end system is able to process images at more than 8 frames per second on a regular laptop computer. Quantitative and qualitative tests in natural conditions have shown promising results in terms of robustness and accuracy.

The main shortcoming of the proposed system is its reduced accuracy in the vertical component of the estimated gaze point. Future work will explore strategies to overcome



(a) Visual exploration of three users in a simple rectangular sequence (blue line).

(b) Visual exploration of three users in a complex video sequence (blue line).

Figure 10: Hit maps of the three users involved in the experiment. For color scale, lightest colors refer to the beginning of the visual explorative session whereas darkest colors refer to the final part of the video.

this problem, as well as methods to automatically calibrate some of the user-dependent system parameters (e.g. interpupillary distance). Finally, we plan to benchmark our system against existing data sets such as MPIIGaze [33].

Acknowledgement: D. Cazzato has been partially supported by “The Foundation BLANCEFLOR Boncompagni Ludovisi, née Bildt”. R. Manduchi has been supported by Reader’s Digest/Research to Prevent Blindness.

References

[1] K. Lund, The importance of gaze and gesture in interactive multimodal explanation, Language Resources and Evaluation, 2007, 41(3-4), 289–303

[2] J. De Villiers, The interface of language and theory of mind, Lingua, 2007, 117(11), 1858–1878

[3] http://www.tobii.com [Online; accessed 01-December-2017]

[4] http://www.sr-research.com/ [Online; accessed 01-December-2017]

[5] A. Duchowski, Eye tracking methodology: Theory and practice, Springer Science & Business Media, 2007, 373

[6] C. H. Morimoto, M. R. Mimica, Eye gaze tracking techniques for interactive applications, Computer vision and image understanding, 2005, 98(1), 4–24

[7] D. W. Hansen, Q. Ji, In the eye of the beholder: A survey of models for eyes and gaze, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(3), 478–500

[8] M. A. Just, P. A. Carpenter, Eye fixations and cognitive processes, Cognitive psychology, 1976, 8(4), 441–480

[9] J. H. Goldberg, M. J. Stimson, M. Lewenstein, N. Scott, A. M. Wichansky, Eye tracking in web search tasks: design implications, in Proceedings of the 2002 symposium on Eye tracking research & applications, ACM, 2002, 51–58

[10] P. Majaranta, A. Bulling, Eye tracking and eye-based human–computer interaction, in Advances in Physiological Computing, Springer, 2014, 39–65

[11] K. Yun, Y. Peng, D. Samaras, G. J. Zelinsky, T. L. Berg, Exploring the role of gaze behavior and object detection in scene understanding, Frontiers in psychology, 2013, 4(no. DEC)

[12] T. Busjahn, R. Bednarik, C. Schulte, What influences dwell time during source code reading?: analysis of element type and frequency as factors, in Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, 2014, 335–338

[13] H. H. Greene, K. Rayner, Eye movements and familiarity effects in visual search, Vision research, 2001, 41(27), 3763–3773

[14] P. Kasprowski, O. V. Komogortsev, A. Karpov, First eye movement verification and identification competition at BTAS 2012, in IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS 2012), 2012, 195–202

[15] F. Deravi, S. P. Guness, Gaze trajectory as a biometric modality, in BIOSIGNALS, 2011, 335–341

[16] M. Wedel, R. Pieters, Eye tracking for visual marketing, Now Publishers Inc, 2008

[17] K. Gidlöf, A. Wallin, R. Dewhurst, K. Holmqvist, Using eye tracking to trace a cognitive process: Gaze behaviour during decision making in a natural environment, Journal of Eye Movement Research, 2013, 6(1), 1–14

[18] H. Cai, X. Zhou, H. Yu, H. Liu, Gaze estimation driven solution for interacting children with ASD, in 2015 International Symposium on Micro-NanoMechatronics and Human Science (MHS), 2015, 1–6

[19] S. Thill, C. A. Pop, T. Belpaeme, T. Ziemke, B. Vanderborght, Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook, Paladyn, 2012, 3(4), 209–217

[20] S. Sheikhi, J.-M. Odobez, Combining dynamic head pose–gaze mapping with the robot conversational state for attention re-

cognition in human–robot interactions, Pattern Recognition Letters, 2015, 66, 81–90

[21] M. P. Michalowski, S. Sabanovic, R. Simmons, A spatial model of engagement for a social robot, 9th IEEE International Workshop on Advanced Motion Control, IEEE, 2006, 762–767

[22] T. Yonezawa, H. Yamazoe, A. Utsumi, S. Abe, Attractive, informative, and communicative robot system on guide plate as an attendant with awareness of user's gaze, Paladyn, Journal of Behavioral Robotics, 2013, 4(2), 113–122

[23] S. Frintrop, Towards attentive robots, Paladyn, Journal of Behavioral Robotics, 2011, 2(2), 64–70

[24] M. Leo, G. Medioni, M. Trivedi, T. Kanade, G. M. Farinella, Computer vision for assistive technologies, Computer Vision and Image Understanding, 2017, 154, 1–15

[25] E. D. Guestrin, M. Eizenman, General theory of remote gaze estimation using the pupil center and corneal reflections, IEEE Transactions on biomedical engineering, 2006, 53(6), 1124–1133

[26] H. Yamazoe, A. Utsumi, T. Yonezawa, S. Abe, Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions, in Proceedings of the 2008 symposium on Eye tracking research & applications, ACM, 2008, 245–250

[27] E. Wood, A. Bulling, Eyetab: Model-based gaze estimation on unmodified tablet computers, in Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, 2014, 207–210

[28] L. Sun, Z. Liu, M.-T. Sun, Real time gaze estimation with a consumer depth camera, Information Sciences, 2015, 320, 346–360

[29] D. Cazzato, A. Evangelista, M. Leo, P. Carcagně, C. Distante, A low-cost and calibration-free gaze estimator for soft biometrics: An explorative study, Pattern Recognition Letters, 2015

[30] X. Xiong, Z. Liu, Q. Cai, Z. Zhang, Eye gaze tracking using an RGBD camera: a comparison with a RGB solution, in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM, 2014, 1113–1121

[31] L. Jianfeng, L. Shigang, Eye-model-based gaze estimation by RGB-D camera, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, 592–596

[32] Z. Guo, Q. Zhou, Z. Liu, Appearance-based gaze estimation under slight head motion, Multimedia Tools and Applications, 2016, 1–20

[33] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Appearance based gaze estimation in the wild, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, 4511–4520

[34] F. Lu, Y. Sugano, T. Okabe, Y. Sato, Adaptive linear regression for appearance-based gaze estimation, IEEE transactions on pattern analysis and machine intelligence, 2014, 36(10), 2033–2046

[35] O. Williams, A. Blake, R. Cipolla, Sparse and semi supervised visual mapping with the s^ 3gp, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2006, 1, 230–237

[36] K. Liang, Y. Chahir, M. Molina, C. Tijus, F. Jouen, Appearance-based gaze tracking with spectral clustering and semi-supervised Gaussian process regression, in Proceedings of the 2013 Conference on Eye Tracking South Africa, ACM, 2013, 17–23

[37] T. Schneider, B. Schauerte, R. Stiefelhagen, Manifold alignment for person independent appearance-based gaze estimation, in 22nd International Conference on Pattern Recognition (ICPR), IEEE, 2014, 1167–1172

[38] O. Ferhat, A. Llanza, F. Vilariño, A feature-based gaze estimation algorithm for natural light scenarios, in Pattern Recognition and Image Analysis, Springer, 2015, 569–576

[39] P. Koutras, P. Maragos, Estimation of eye gaze direction angles based on active appearance models, IEEE International Conference on Image Processing (ICIP), IEEE, 2015, 2424–2428

[40] H. Yoshimura, M. Hori, T. Shimizu, Y. Iwai, Appearance based gaze estimation for digital signage considering head pose, International Journal of Machine Learning and Computing, 2015, 5(6), 507

[41] F. Lu, Y. Sugano, T. Okabe, Y. Sato, Gaze estimation from eye appearance: A head pose-free method via eye image synthesis, IEEE Transactions on Image Processing, 2015, 24(11), 3680–3693

[42] Y. Sugano, Y. Matsushita, Y. Sato, Learning-by-synthesis for appearance-based 3d gaze estimation, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, 1821–1828

[43] C. Xiong, L. Huang, C. Liu, Remote gaze estimation based on 3d face structure and iris centers under natural light, Multimedia Tools and Applications, 2015, 1–15

[44] K. A. Funes-Mora, J.-M. Odobez, Gaze estimation in the 3d space using RGB-D sensors, International Journal of Computer Vision, 2016, 118(2), 194–216

[45] F. Lu, T. Okabe, Y. Sugano, Y. Sato, Learning gaze biases with head motion for head pose-free gaze estimation, Image and Vision Computing, 2014, 32(3), 169–179

[46] C. Holland, A. Garza, E. Kurtova, J. Cruz, O. Komogortsev, Usability evaluation of eye tracking on an unmodified common tablet, in CHI'13 Extended Abstracts on Human Factors in Computing Systems, ACM, 2013, 295–300

[47] J. Chen, Q. Ji, A probabilistic approach to online eye gaze tracking without explicit personal calibration, IEEE Transactions on Image Processing, 2015, 24(3), 1076–1086

[48] F. de la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, J. Cohn, Intraface, 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), IEEE, 2015, 1, 1–8

[49] M. Smereka, I. Duleba, Circular object detection using a modified Hough transform, International Journal of Applied Mathematics and Computer Science, 2008, 18(1), 85–91

[50] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2001, 1, 1–511

[51] X. Xiong, F. Torre, Supervised descent method and its applications to face alignment, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, 532–539

[52] N. A. Dodgson, Variation and extrema of human interpupillary distance, in Electronic imaging, 2004, 36–46, International Society for Optics and Photonics, 2004

[53] C. C. Gordon, C. L. Blackwell, B. Bradtmiller, J. L. Parham, P. Barrientos, S. P. Paquette, B. D. Corner, J. M. Carson, J. C. Venezia, B. M. Rockwell, et al., 2012 anthropometric survey of us army personnel: Methods and summary statistics, tech. rep., Army Natick Soldier Research Development And Engineering Center Ma,

2014

[54] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 6, 679–698

[55] J. F. Magee, Decision trees for decision making, Harvard Business Review, 1964

[56] G. Fanelli, M. Dantone, J. Gall, A. Fossati, L. Van Gool, Random forests for real time 3d face analysis, International Journal of Computer Vision, 2013, 101(3), 437–458

[57] L. Breiman, Random forests, Machine learning, 2001, 45(1), 5–32

[58] A. Liaw, M. Wiener, Classification and regression by randomforest, R news, 2002, 2(3), 18–22

[59] G. Bradski, A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, O'Reilly Media, Inc., 2008

[60] https://www.qt.io/developers [Online; accessed 01-December-2017]

[61] M. Leo, D. Cazzato, T. DeMarco, C. Distante, Unsupervised eye pupil localization through differential geometry and local self-similarity matching, PloS one, 2014, 9(8), e102829

[62] R. Valenti, T. Gevers, Accurate eye center location through invariant isocentric patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(9), 1785–1798

[63] M. Leo, D. Cazzato, T. DeMarco, C. Distante, Unsupervised approach for the accurate localization of the pupils in near frontal facial images, Journal of Electronic Imaging, 2013, 22(3), 033033–033033

[64] S. Asteriadis, P. Tzouveli, K. Karpouzis, S. Kollias, Estimation of behavioral user state based on eye gaze and head pose—application in an e-learning environment, Multimedia Tools and Applications, 2009, 41(3), 469–493

[65] T. D'Orazio, M. Leo, C. Guaragnella, A. Distante, A visual approach for driver inattention detection, Pattern Recognition, 2007, 40(8), 2341–2355

[66] V. Sundstedt, Gazing at games: An introduction to eye tracking control, Synthesis Lectures on Computer Graphics and Animation, 2012, 5(1), 1–113

[67] L. Chaby, M. Chetouani, M. Plaza, D. Cohen, Exploring multimodal social-emotional behaviors in autism spectrum disorders: an interface between social signal processing and psychopathology, in International Conference on Privacy, Security, Risk and Trust (PASSAT), and International Confernece on Social Computing (SocialCom), IEEE, 2012, 950–954

[68] L. Piccardi, B. Noris, O. Barbey, A. Billard, G. Schiavone, F. Keller, C. von Hofsten, Wearcam: A head mounted wireless camera for monitoring gaze attention and for the diagnosis of developmental disorders in young children, in the 16th IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), IEEE, 2007, 594–598

[69] X. Li, A. Çöltekin, M.-J. Kraak, Visual exploration of eye movement data using the space-time-cube, in International Conference on Geographic Information Science, Springer, 2010, 295–309

[70] A. T. Duchowski, V. Shivashankaraiah, T. Rawls, A. K. Gramopadhye, B. J. Melloy, B. Kanki, Binocular eye tracking in virtual reality for inspection training, in Proceedings of the symposium on Eye tracking research & applications, ACM, 2000, 89–96