journal homepage: www.intl.elsevierhealth.com/journals/cmpb

A neural-based remote eye gaze tracker under natural head motion☆

*Diego Torricelli**, *Silvia Conforto, Maurizio Schmid, Tommaso D'Alessio*

Department of Applied Electronics, University Roma TRE, Italy

ARTICLE INFO

Article history:
Received 16 September 2007
Received in revised form 6 June 2008
Accepted 8 June 2008

Keywords:
Eye gaze tracking
Human machine interaction
Neural networks
Eye detection

ABSTRACT

A novel approach to view-based eye gaze tracking for human computer interface (HCI) is presented. The proposed method combines different techniques to address the problems of head motion, illumination and usability in the framework of low cost applications. Feature detection and tracking algorithms have been designed to obtain an automatic setup and strengthen the robustness to light conditions. An extensive analysis of neural solutions has been performed to deal with the non-linearity associated with gaze mapping under free-head conditions. No specific hardware, such as infrared illumination or high-resolution cameras, is needed, rather a simple commercial webcam working in visible light spectrum suffices. The system is able to classify the gaze direction of the user over a 15-zone graphical interface, with a success rate of 95% and a global accuracy of around 2°, comparable with the vast majority of existing remote gaze trackers.

1. Introduction

Eye gaze tracking (EGT) estimates the direction of gaze of a person [1]. In biomedicine, estimating the direction of gaze is of utmost importance not only in diagnostic contexts, but also in the framework of assistive technology to facilitate interaction with interfaces for people with disabilities [2]. To accomplish this task, different approaches have been proposed over the last three decades [3]. Most of them have been proven as accurate, and some of the techniques and systems gave rise to commercially available products [4–7].

To enhance the accuracy of these techniques, the most common solution is based on the presence of intrusive devices, such as chin rests to restrict head motion, and some of them need the user to wear equipment, such as *ad hoc* contact lenses [8], electrodes [9] or head-mounted cameras [10].

The usability requirements that an ideal EGT system should satisfy to be applied as an assistive technology device in the interaction with general computer interfaces [11] are listed below and will focus our attention during the discussion of this paper:

- tracking accuracy and reliability;
- robustness to light conditions and head movements;
- non-intrusiveness (i.e. cause no harm or discomfort);
- real-time implementation;
- minimal burden time for calibration;
- cost-effectiveness.

Among all the EGT systems, remote eye gaze trackers (REGT) represent an effective non-intrusive solution and are thus becoming prevalent in the context of assistive

☆ *Disclosure*: All the subjects who performed the trials gave their informed consent to the experiments, according to the Helsinki declaration, and the local committee approved the experimental procedure.

* *Corresponding author at*: Department of Applied Electronics, University Roma TRE, Via della Vasca Navale 84, I-00146 Roma, Italy. Tel.: +39 06 5733 7298; fax: +39 06 5733 7026.

E-mail address: d.torricelli@uniroma3.it (D. Torricelli).

0169-2607/$ – see front matter © 2008 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2008.06.008

technology. With these systems, no devices are in physical contact with the user. It is possible to estimate the gaze direction by tracking some typical visual features of the eye.

Starting from an image sequence (acquired either by a single camera or by a set of multiple cameras), a preliminary task has to be accomplished, that is the detection of face and eyes within the image frames. Some local features of the eyes are then needed to estimate the position of the pupil with respect to the eye socket. Based on this information, a mapping between the eye configuration and the direction of gaze can be computed.

Following the aim of providing an effective solution to this issue, the proposed work presents a novel approach to view-based REGT systems, which is based on neural computing and commercially available devices. The system, developed to target people with disabilities, was tested in terms of robustness to head motion and changes in light conditions.

The following section describes previous REGT research. In Section 3 the proposed method will be introduced and described in detail. In Section 4 the experimental setup and testing will be presented. Finally, the experimental results will be discussed in Sections 5 and 6.

2. Background

Two classes of approaches exist among the REGT systems: one widespread and effective solution is based on active infrared (IR) illumination [12–15]; the second relies on the analysis of videos capturing eye movements under natural light conditions, and is commonly referred to as view-based or appearance-based [16–18]. Both approaches are described in the following.

2.1. *Infrared-based REGT*

The IR-based techniques utilize active illumination from infrared light sources to enhance the contrast between the pupil and the iris. The light beam produces two effects on the eye. The first one, called bright-eye effect, is similar to the red-eye effect in photography: the camera records a bright circle, which is the light passing through the pupil and reflected by the retina. The second effect is the reflection of the light on the corneal surface (the external surface of the eye), seen by the camera as a small bright point called “glint” or “corneal reflection”. Assuming that the eye is a sphere that rotates around its centre, the glint does not move with the eye rotation, whereas the pupil does. For this reason the glint can be considered as a reference point.

After grabbing the eye image, pupil and glint can be detected by appropriate image processing algorithms [4,19]. Gaze detection is then computed by applying a mapping function that, starting from the 2D pupil-glint vector, calculates the point observed on the screen. To this latter aim, different mapping functions have been proposed [1,20,21]. Among them, the most commonly used [13] is based on a second order polynomial function defined as

$$\begin{cases} s_x = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \\ s_y = b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + b_5 y^2 \end{cases} \quad (1)$$

where (s_x, s_y) are the screen coordinates, and (x, y) are the pupil-glint vector components. A calibration procedure is thus needed to estimate the 12 unknown variables $\{a_0, \ldots, a_5, b_0, \ldots, b_5\}$. To perform the calibration, in Ref. [13] the user is requested to look at nine points on the screen. Since each calibration point defines two equations, the system is over constrained with 12 unknowns and 18 equations, and can be solved, e.g. through least squares analysis.

The accuracy of the aforementioned IR-based methods corresponds to a standard deviation of about 1–1.5°, which represents a good figure for the requirements of the vast majority of the interactive applications. Moreover, the calibration is relatively fast and easy, and the required computational time allows real-time interaction. Thus, IR-based REGT systems are prevalent in both research and commercial environment. Nevertheless, besides the high cost, several important issues have yet to be solved, and at present hinder IR-based REGT systems usability:

- Changes in *light conditions*, especially in presence of sun light, which interferes with the infrared illumination, generating other kinds of reflective phenomena [1,3]. Some solutions have been proposed for this issue [19].
- *Bright eye effect variability* in different subjects. This effect has been verified as preventing IR-based techniques from providing consistent and reliable results in gaze direction estimation [4].

Moreover, there is agreement that EGT systems may find use as future computer input devices only if they become convenient and inexpensive [22]. This is one of the reasons why a different class of REGT systems has become more frequent in research literature to meet these requirements, i.e. view-based REGT. These will be described in the following.

2.2. *View-based REGT*

In the classical view-based approaches, intensity images of the eyes are grabbed by traditional image acquisition devices and then processed to extract information on the eye configuration. No *ad hoc* hardware is usually needed. On the other hand, more challenging efforts are required in terms of image processing, as compared to the IR-based REGT systems, in order to detect face and eyes.

In the work by Baluja and Pomerleau [16] no explicit features are required. Each pixel of the image is considered as an input parameter to the mapping function. A 15 × 40 pixels image of the eye is processed, corresponding to a 600-dimension input space vector. An artificial neural network (ANN), trained to model the relationship between the pixel values and the 2D coordinates of the observed point on the screen, is used as the mapping function. In the calibration procedure, the user is requested to look at a cursor moving on the screen along a known path made of 2000 positions. An accuracy of 1.5° is reported.

Similar to the previous work, Xu et al. [18] present a neural technique as well. The image of the eye is segmented to precisely locate the eye and then processed in terms of histogram normalization to enhance the contrast between the eye features. As in Baluja and Pomerleau's method, the ANN receives 600-image pixel values and returns a vector of possible (*x*, *y*) coordinates of the estimated gaze point with an accuracy of around 1.5°. Three thousand examples are used for the training procedure.

Tan et al. [17] proposed a method that considers the image as a point in a multi-dimensional space, by using an appearance-manifold technique: an image of 20×20 pixels can be considered as a 400-component vector in a 400-dimensional space. In this work, 252 images from three different subjects were taken, using each one as the test and the others as the manifold. The calibration is done by looking at markers on a screen, while taking the picture of the eye for each position. The reported accuracy, as calculated by the leave-one-out approach, is very good (0.38°).

Zhu and Yang [21] propose a method for gaze estimation based on feature extraction from an intensity image. Both the irises and the inner corners of the eye are extracted and tracked with sub-pixel accuracy. Then, through a linear mapping function, they calculate the gaze angle, using only two points for the calibration. The reported accuracy is 1.4°.

2.3. *Head motion*

Compensating for head motion still represents the greatest limitation for most of the remote eye gaze trackers [23]. In principle, as the head moves from its original position, a new calibration is needed. Thus, a strong requirement for the system to work properly is to maintain the head perfectly still. At the same time, this requirement is by far very restrictive for most of the applications. Several researchers over the last years have dedicated their efforts to the solution of this issue, with the goal of allowing the user to move the head freely, yet maintaining the accuracy in an acceptable range.

To compensate for head movements, most of the related methods are based on a 3D model of the eye [12,15,24–26]. In such methods, two or more cameras are used to estimate the 3D position of the eye. Through the use of mathematical transformations, the line of gaze is then computed. In some of these systems [13,24], pan and tilt cameras with zoom lenses are required to follow the eye during head movements. An original approach proposed by Yoo and Chung [15] uses a multiple-light source that gives rise to a calibration-free method. Accurate results are also obtained by Park [27], which uses a three-camera system to compensate for head motion.

Even if these approaches seem to solve the problem of head motion, the complexity of the proposed systems, driven by the need of additional hardware, prevents them from being routinely used in a home environment. To accommodate this need, Ji and Zhu proposed a fairly simple method to compensate for head motion, based on artificial neural networks (ANN), and a single IR-based camera [20]. They criticize the linear/quadratic mapping functions because they cannot take into account the perspective projection and the orientation of the head. Hence, they proposed a different set of eye features: beside the pupil-glint vector, other four factors are chosen for the gaze calibration to get the mapping function. A generalized regression neural network (GRNN) is then used to accomplish the task, by using a normalized input vector comprising the six factors. The GRNN can classify one of eight screen regions with a success of 95%.

Solving the issue related to head motion is thus common to both IR-based and view-based REGT systems: if it is likely that this problem is close to the solution for the first ones, work still needs to be done in view-based systems. Even if Baluja and Pomerleau's technique allows for small head movements, the only approach specifically addressed to solve this issue has been recently investigated by the authors of the present work [28]. In the cited work, a multilayer perceptron has been trained to account for little head movements that occur when the user is asked to maintain the head still and look to a cursor on the screen. The encouraging results persuaded us to deepen the field of the neural approach for robust gaze tracking within less constrained conditions.

2.4. *Design remarks/considerations*

The presented eye-gaze tracker has been designed to meet most of the requirements mentioned as of utmost importance in Section 1. In particular, the design has been focused on the robustness to light conditions and head pose. A procedure for an automatic initialization is also proposed. In the following section the method will be presented and described in detail.

3. System description

As displayed in Fig. 1, the procedure for the estimation of the gaze is composed of two blocks: the first one makes use of image processing algorithms to extract and track the features of the eyes; the second block accomplishes the task of mapping the geometric visual information with the gaze direction, by using artificial neural networks. Merging these two techniques has a double purpose: on the one hand to both guarantee robustness to light changes and minimize user intervention in the init process. On the other hand the neural approach has the goal of account for the high non-linearity of the mapping function. To the authors' knowledge

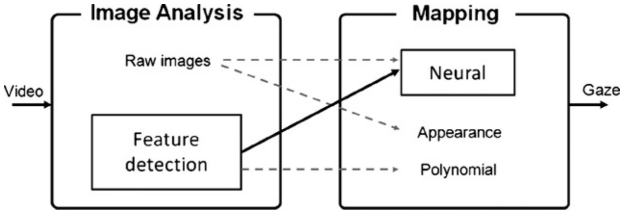

Fig. 1 – General scheme of a procedure for a view-based REGT system. The dashed arrows indicate the classical view-based techniques. The solid arrow shows the proposed approach, where feature detection is combined with neural mapping to make the system robust to light changes and head movements.

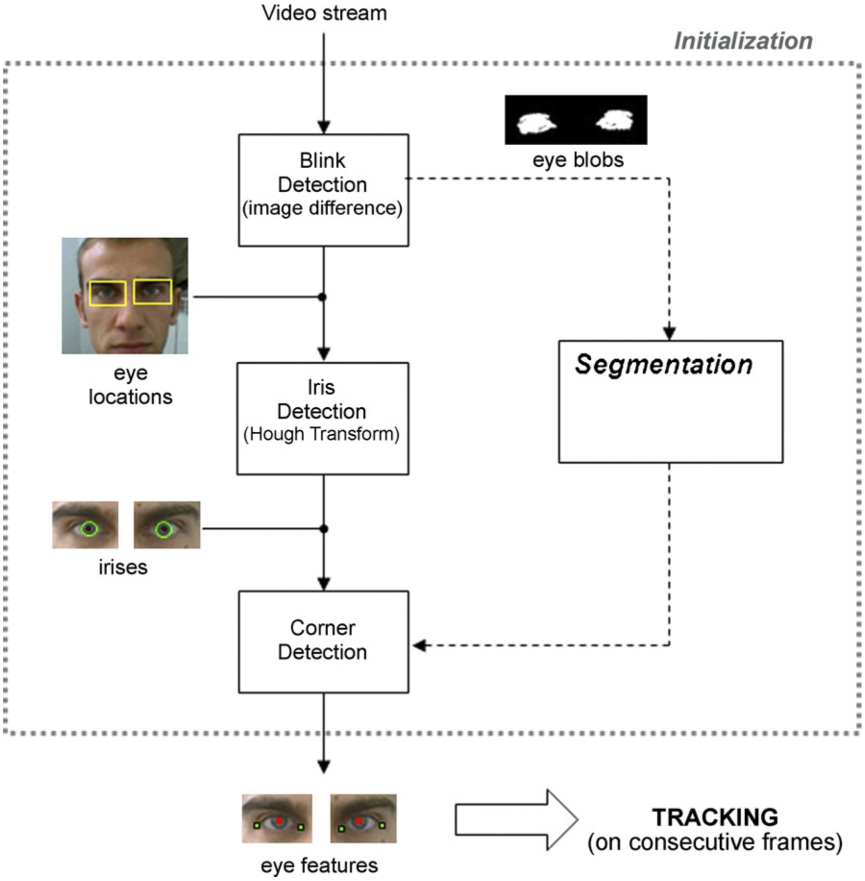
Video stream
Initialization
Blink Detection (image difference)
eye blobs
eye locations
Iris Detection (Hough Transform)
Segmentation
irises
Corner Detection
eye features
TRACKING (on consecutive frames)

Fig. 2 – The initialization procedure. Three consecutive steps have been adopted and integrated in the initialization procedure: blink detection, iris detection and corner detection. This procedure allows the system to extract the feature templates to be used by the subsequent tracking procedure.

this approach has never been presented for view-based REGT systems before.

In the next subsection, the first portion of the proposed REGT procedure will be described and denoted as feature detection and tracking. Then we will focus on the mapping section, which will be denoted as neural mapping.

3.1. *Feature detection and tracking*

To gather information on the configuration of the eyes, two kinds of features are needed, i.e. the pupil and one or more reference points on the face. Since no infrared light sources have been used, such reference points could not be represented by corneal reflections. As it will be clarified later, two reference points per-eye have been considered, i.e. the internal and external corners of the eye socket. Both eyes are taken into account, to allow for redundancy in the mapping function, thus providing a set of six eye features.

Starting from an image sequence, three steps are performed, as displayed in Fig. 2. With a preliminary blink detection procedure, eye locations are estimated through image differencing. Irises are then detected with the use of a modified Hough transform algorithm [29]. At last, a corner detection block extracts the corner positions by analysing and segmenting the eye-blobs resulting from the blink detection module. This procedure permits to automatically initiate the tracking process, based on template matching.

3.1.1. *Blink detection*

This block is mainly based on the analysis of eyelid movements, in order to identify the shape of the eye as the area of the image where high changes of gray level intensity occur.

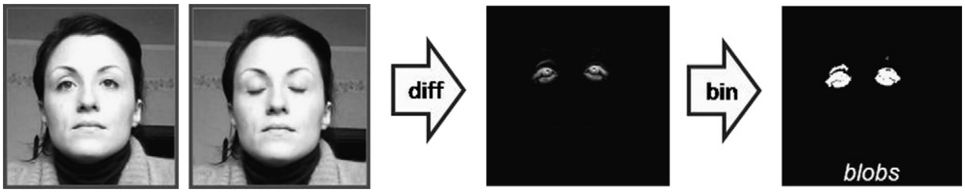

Fig. 3 – Blink detection. The image difference between two consecutive frames is followed by a gray-to-black and white conversion to detect the presence of a blink.

The user is just requested to blink in front of the camera. The algorithm then performs both a frame differencing between couples of subsequent frames, and a gray-to-binary conversion. The resulting binary images (see Fig. 3) show the zones with high changes in pixel value ("blobs" in the following).

A five-step sequence is then carried out to verify that each blob actually represents the eyes:

(i) Erode and dilate to account for fragmentation of the eye blobs.
(ii) Compute the number of clusters: only the images with two clusters can be considered as eye-candidates.
(iii) Calculate the ratio between the size of the two clusters and the distance between them to account for the anatomy of the face, and to make the technique independent from the camera-user distance.
(iv) Calculate the inclination of the line passing through the centroids of the clusters. A ±30° range is allowed to account for a natural head pose.
(v) Delete false positives. It is a post-processing step that aims at removing such images that contain estimated eyes with a size too different from the size of the eye candidates present in the other images.

This sequence of steps will lead to the definition of a set of binary images containing the shapes of the eyes. This set of images will represent the basis of the next two blocks, i.e. iris and corner detection.

3.1.2. *Iris detection*

Once each eye has been detected from the previous step, the gray intensity images of the eyes are extracted from the last frame of the initialization video and then processed by a two step algorithm, constituted of:

- *Edge detection*: Based on the observation that the iris and the sclera present high contrast in brightness, the edge detection can easily serve as a detector of the iris. Different kinds of edge detectors have been evaluated. Among different image frame operators (see e.g. [30,31]) the Sobel operator accomplished the task with a higher specificity in terms of border detection: even if Canny operator results to be in general more robust to light changes, a very high number of edges is detected within the eye image, making the discrimination of the correct iris edge very difficult. With the Sobel operator a lower number of edges is detected. Nevertheless, thank to its high contrast, the iris edge always belongs to the detected borders, making it possible to automatically choose the correct iris border with Hough transform technique, as explained in the following.
- *Modified Hough transform*. The Hough transform is a feature extraction technique aimed at finding a certain class of shapes, such as lines, circles and ellipses, within an image [29]. In the case of circumferences, for a given value of radius, the algorithm will find the best "circumference candidates". A voting number is assigned to each candidate, representing the number of pixels of the image that belongs to that circumference. Since the exact value of the radius is unknown, the algorithm has been applied iteratively for different radius values around a first guess value obtained automatically by dividing the inter-eyes distance by a constant factor (coming from morphological observations). To estimate the correct pair of circumferences from the set of candidates, a modification of the algorithm has been applied. Each group of similar circumferences, in terms of radius and centre position, is merged into a new circumference and a new vote is assigned to it, coming from a weighted sum of the single ones. 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 (Fig. 4). This process has shown a good behaviour over different light conditions even with asymmetric illumination of the face.

3.1.3. *Corner detection*

To determine the eye corners, a further processing of the results of the procedure of blink detection is needed. As mentioned, the frame-to-frame difference images highlight the zones in which a movement occurs, based on the fact that during a movement the corresponding pixel of the image changes its value. We are interested in getting the shape of the eye as accurately as possible by separating the zones with large inter-frame motion (eyelids) from those with small one. To choose the appropriate threshold value in an automatic way, the difference images, each one with its absolute values, are

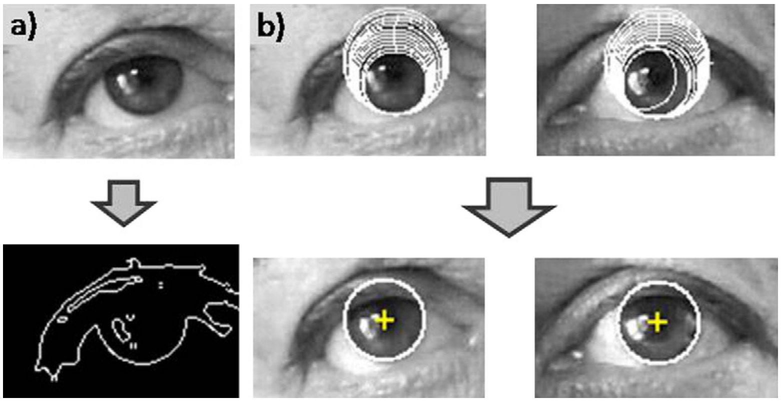

Fig. 4 – Iris detection. The Sobel operator is applied to detect the edges in the image (a) and a modified Hough transform algorithm is then used to choose the two most voted circumferences among the candidates (b).

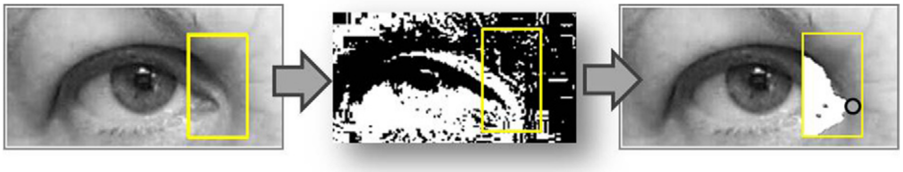

Fig. 5 – Inner corner detection. From the knowledge of the position of the irises, a search window is created. Then the blobs coming from the blink detection block are filtered to detect the inner extreme of the eyelid movements, i.e. the inner corner of the eye.

summed. The resulting matrix is then filtered to determine the eye corner, as explained in the following.

For the inner corner, a search window is generated, not including the iris. Within the window, the mean value and the standard deviation of the values of the image are used to define the threshold for the image binary process. As shown in Fig. 5 the most lateral pixel of the binary image is considered as the estimated inner corner.

For the external corner a search window is created over the external area of the eye, starting from the estimated iris position. Ten-level quantization is then applied to the intensity image. By eliminating the brighter levels, the line of the upper eyelid can be easily identified. The external extremity of this shape will be considered as the external corner (see Fig. 6). In the experimental testing section, the results of this technique, over different light conditions and distance of the user from the camera, will be shown.

3.1.4. *Feature tracking*

The initialization process above discussed has the aim of automatically detecting the features of the eyes, thus avoiding the presence of manual selection that could compromise the accuracy, repeatability, speed and convenience of the setup. The detected features represent the starting point for the tracking procedure, which is based on a hierarchical procedure combining template matching [32] and sub-pixel approximation [33].

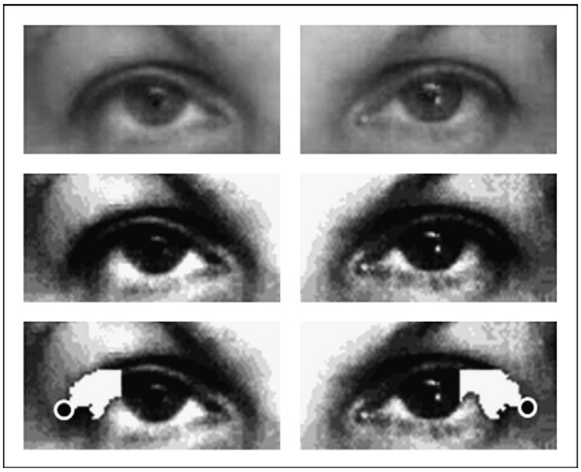

Fig. 6 – External corner detection. The gray level image of the eyes is enhanced using histogram equalization. A binary image (containing the white areas) is created by eliminating the pixels with higher luminance. The external extremes are then taken as the external corners of the eyes.

3.2. *Neural mapping*

The second part of the REGT procedure is based on the use of artificial neural networks designed to map the relation between the eye features and the gaze direction. The neural approach has been chosen for the ability of the nets to learn by examples, to smoothly adapt to changing contexts and to generalize for unseen configurations. At the same time, due to the underlying complexity, the neural architecture requires an accurate design that regards the choice of the typology of the input, and the selection of the internal structure of the net, i.e. the number of hidden layers and neurons. The proposed approach arises from the belief that, once provided with a proper set of input space vectors and training examples, the net will be able to generalize for different head positions. In the next paragraphs the chosen input set, the different nets and the training procedures will be detailed.

3.2.1. *Input space selection*

The choice of the input set has the role of giving the net the appropriate information about the relative position of the eye within the eye-socket like on the pose of the head with respect to the screen. This information is combined by the net to build a non-linear regression function that takes into account those head movements that can occur during the visual task. Basically two features are needed: one refers to the pupil, and the other one, called reference point, to appropriately describe the movement of the head. The latter has been identified as the corner of the eye.

As opposed to what happens with infrared-based techniques, the image of the eye in visible light spectrum does not permit to identify a reference point on the surface of the eye. Whereas the “glint” of IR-based systems is pretty insensitive to the rotation of the eyes around its optic axis, the vector connecting the eye corner with the pupil turns out to be extremely sensitive to small head rotations. For this reason, and for the unavoidable uncertainty of feature tracking with low-resolution images, a redundancy has been introduced by considering, for each eye, two vectors connecting, respectively, the pupil with the inner and the external corner. The resulting vector of inputs thus consists of 12 parameters: eight of them come from the magnitudes and angles of the distance vec-

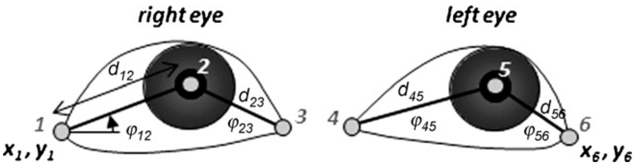

Fig. 7 – Geometric features of the eyes (numbered from 1 to 6). Magnitudes and angles of the distance vectors represent 8 of the 12 inputs. The remaining 4 inputs are the x and y coordinates of the external corners of the eyelids. For the sake of clarity, only one of the angle features is represented.

tors previously mentioned; the remaining four consist of the *x* and *y* coordinates of the two external corners. All of them are depicted in Fig. 7.

3.2.2. *The neural structure*

Two different architectures of neural networks have been used to explore and approximate the mapping properties of the gaze function. The first net, a multilayer perceptron (MLP) [34], is considered to have a strong ability to provide compact and robust representation of mapping in real-world problems. The other net is represented by a general regression neural network (GRNN) [35], which is recognized to have better performance in function approximation tasks with respect to traditional neural networks. A GRNN has also been used in one previous work [14], based on infrared-based techniques. Both networks have a 12-neuron input layer receiving the vector of the eye parameters. Each architecture consists of two distinct nets that separately calculate the x_p and y_p coordinates of the screen point.

The MLP design aims at finding the optimum configuration with the following variables: number of hidden layers, number of neurons, number and typology of training trials and number of training epochs. To evaluate the best configuration, accuracy is taken as the performance index. The structures of the chosen MLP are displayed in Fig. 8.

The GRNN, based on the radial basis architecture, consists of an input layer, a hidden layer, a summation layer and an output layer (see Fig. 8). The hidden layer has as many neurons as there are input/target vectors, while the number of nodes of the summation layer equals the number of output neurons plus one. The hidden layer contains a parameter called *spread* that determines the width of an area in the input space to which each neuron responds: the higher the spread, the smoother the function. The choice of the best configuration lies basically in the optimal estimation of the spread, which constitutes the only user-defined internal parameter.

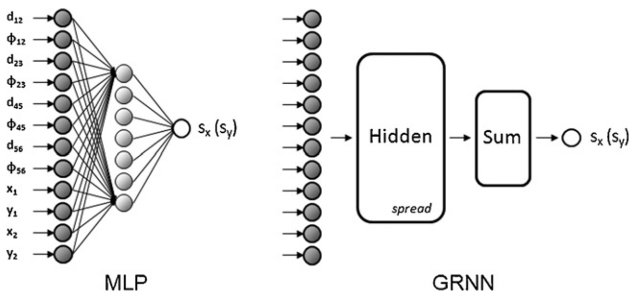

Fig. 8 – The multilayer perceptron (MLP, left) and the general regression network (GRNN, right), both with a 12 input layer and a single output neuron for the calculation of either the x or the y coordinate of the observed point on the screen.

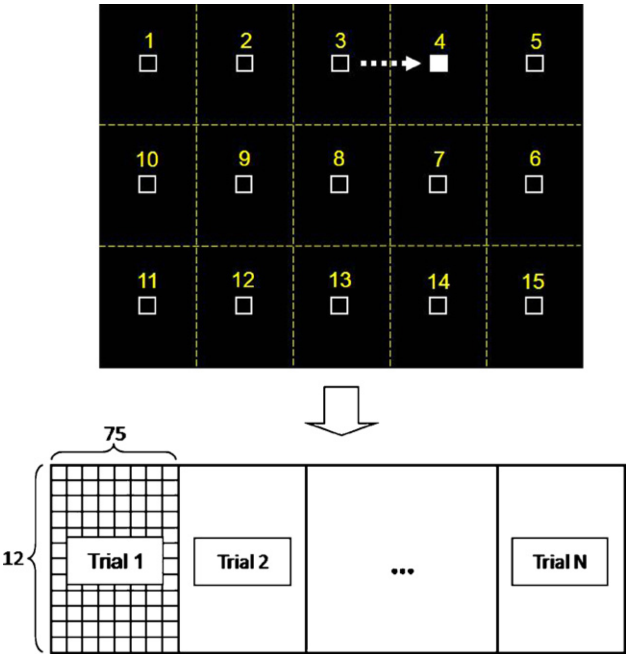

Fig. 9 – The training procedure. The upper panel represents the screen with the 15 positions that the user is asked to look at during one trial. For each position, 5 video frames are processed and analyzed to build up a 75 columns (5x15) matrix, each column representing a 12 parameters vector. The whole training matrix is represented in the lower panel.

3.2.3. *The training procedure*

The input set is composed of examples from different trials with changes in head position. For each trial the subject is requested to look at a white cursor moving on the screen over 15 different positions. During the task, the webcam grabs seven frames for each observed position. Each frame will be then processed to extract the eye features. During the task the subject is requested to maintain the head in a comfortable position, so that natural movements are allowed. Among the seven frames grabbed during each position, five are taken to build up the training set, while the remaining two are used to validate the training procedure. As shown in Fig. 9, the resulting training set is then constituted of 75 (15 positions, 5 frames per position) vectors for each video trial.

The number of positions covered by the cursor has been set at 15 for the following reasons:

(1) HCI interfaces used for people with disability usually contain a reduced number of icons of relatively large

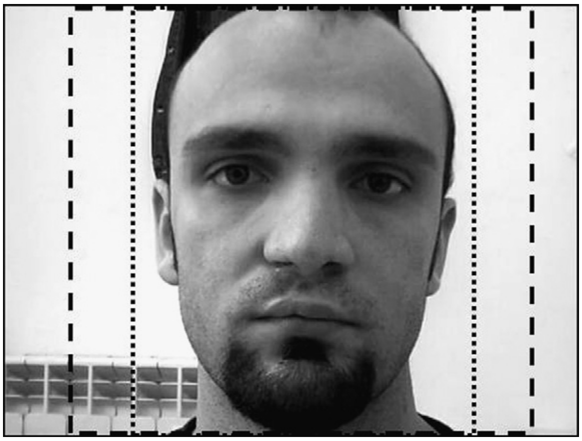

Fig. 10 – Head movements. Dashed line: range of head movement in the training phase. Dotted line: head motion during the test trials.

dimensions, which can roughly be approximated by dividing the screen in 8–15 areas.

(2) A high number of positions leads to time consuming training procedures; increasing the number of areas would unnecessarily increase the burden time for calibration.

To give the ANN the possibility to generalize, a wide range of head displacements is needed. Five head positions have been chosen for this purpose: one in central position, and four in upper-down and left-right positions, as shown in Fig. 10. The input set has been provided together with the desired outputs, represented by the *x* and *y* coordinates of the centre of the cursor. A resilient back propagation training algorithm (RBP) has been chosen. At each epoch of training two kinds of errors have been evaluated and compared each other. The first one is internally calculated by the RBP algorithm at each iteration, and the second one, the *validation* error, is obtained by using the validation set (two frames per position) as input. The end of the training is established by comparing these two values with predefined thresholds, ranging between 10^{-4} and 10^{-3}.

4. Experimental testing

The experimental tests have been carried out to evaluate the accuracy and robustness of the method under conditions that mimic a realistic context of use.

This section has been organized as follows:

- Testing of the algorithms of feature detection and tracking over different subjects, light conditions, and distance to the camera.
- Analysis of different neural configurations: The MLP and the GRNN have been tested by varying the number of neurons and layers in order to set the configurations with the best performance.
- Analysis of the accuracy and robustness of the global system.
- Performance comparison with other methods applicable to the view-based context. In particular a polynomial mapping function [13] has been implemented and tested and the results compared to the neural solution.

The overall hardware composing the system includes a commercial webcam (QuickCam® Communicate STX^{TM}, Logitech®, 30 fps, 640 × 480 pixels), a personal computer with a Pentium-4 3.0 GHz processor, and a 17 in. monitor. The webcam has been located over the central zone on the screen, to minimize the errors coming from perspective distortions.

4.1. *Feature detection and tracking*

A set of separate experiments has been dedicated to validate the initialization technique. Five subjects with age ranging from 22 to 57 years have been analyzed under different conditions of illumination and distance to the camera. For each subject five videos have been analyzed (see Fig. 11): one very close to the camera (25 cm), one pretty far (80 cm) and three videos at a normal distance (40 cm) with different conditions of illumination, high (900 lx), low (250 lx) and asymmetrical, thus resulting in a total number of 25 trials. During the trial the subject is requested to blink three times while looking at the camera.

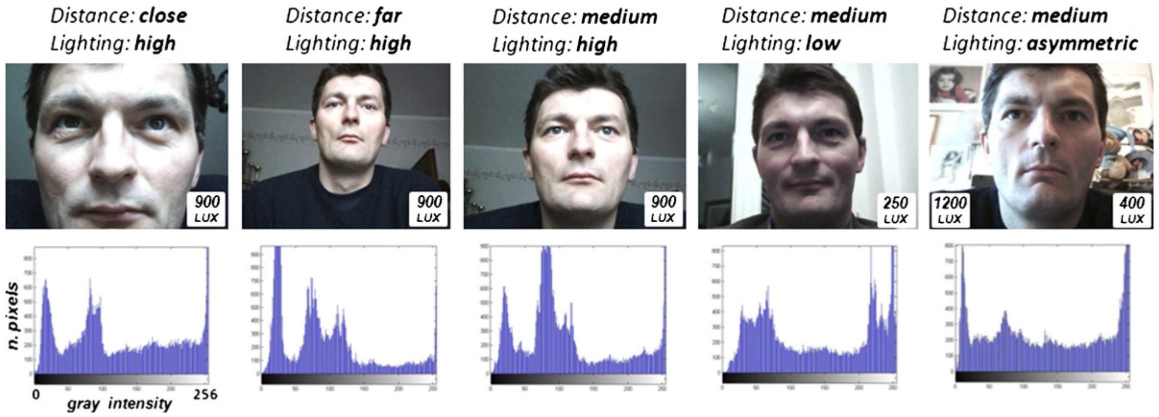

Fig. 11 – Illumination and distance modifications. Five different combinations of illumination and distance from the camera have been tested for the algorithm. Below each picture, the histogram representation of the luminance distribution is depicted.

Table 1 – Feature detection: assigned performance values corresponding to the distance between the automatically and manually detected feature (for a 80 × 40 pixels eye image)

Distance between manual and automatic detection	Performance value	Correctness value (%)
0–1 (pixels)	4	100
2–3	3	90
4–6	2	50
7–9	1	20
>9	0	0

The accuracy has been evaluated in terms of percentage of correct estimation: for each video, a value ranging from 0 to 4 was assigned to each estimated position, based on the distance between the estimated position and the one manually determined by one independent researcher. As an example, considering a 80 × 40 eye image, the relation between performance values and distance ranges is depicted in Table 1. The percentage of correct recognition is reported in Table 2. As illustrated, the method shows a good performance in non-extreme conditions, even for high and low illumination. A strong lateral illumination makes the technique fail in the corner identification.

4.2. *Neural configurations*

In order to detect the most appropriate structure, multiple configurations of MLP and GRNN have been tested. In this part of the study the performance of each net has been evaluated in terms of the ability to classify the gaze direction over the 15 zones on the screen, expressed by the percentage of correctly estimated zones. More than 50 configurations of MLP have been tested, with one, two and three hidden layers and a total number of neurons ranging from 5 to 420.

The experimental protocol has been designed as follows: for each trial session the user is asked (1) to seat in front of the screen, (2) to execute a repetition of three blinks to permit the system to automatically detect the eye features, and then (3) to perform the visual task looking at the cursor moving on the 15-position path. At the end of the trial the subject is asked to stand up and move away from the seat, and then go back again in front of the screen to start a new trial. During the session the subject is free to move the head in a natural way, still avoiding sudden and/or large movements, at a distance from the screen of approximately 46 cm (corresponding to a distance from the camera of 43 cm). The test set includes eight trials with the head approximately centred with respect to the trained volume, and slight movements allowed (Fig. 10). In particular, the head translation lies within an area of 3 × 3 cm in x and y directions, and the variation along z direction is of ±1 cm.

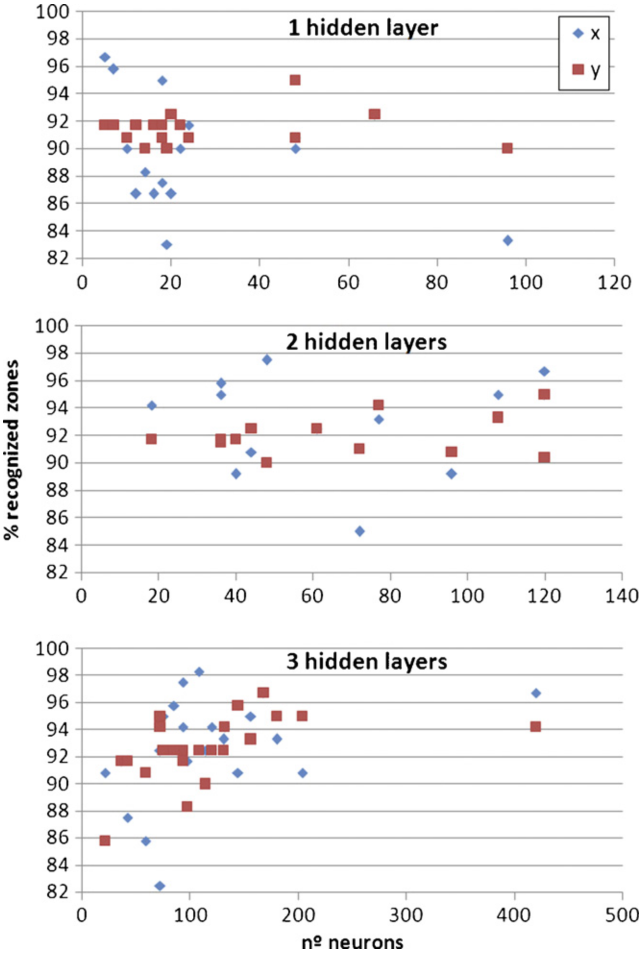

Fig. 12 – MLP classification performance as a function of the number of neurons of the net across the different number of hidden layers.

As shown in Fig. 12, the mean performance does not vary significantly with the number of neurons. A slight improvement (2–3%) occurs in the case of two and three hidden layers. Among the overall set, some nets reach higher recognition percentages. In particular, for the calculation of the *x* coordinates, some configurations achieve a performance of 96–98%, while

Table 2 – Performance of the feature detection algorithm

	Percentage of successful detections			
	Blink (%)	Iris (%)	Inner corner (%)	External corner (%)
Close (25 cm)	100	100	90	100
Far (80 cm)	100	80	100	80
Medium distance (40 cm) (high lighting)	100	90	100	80
Medium distance (40 cm) (low lighting)	100	90	100	100
Medium distance (cm) (asymmetric lighting)	100	90	30	60

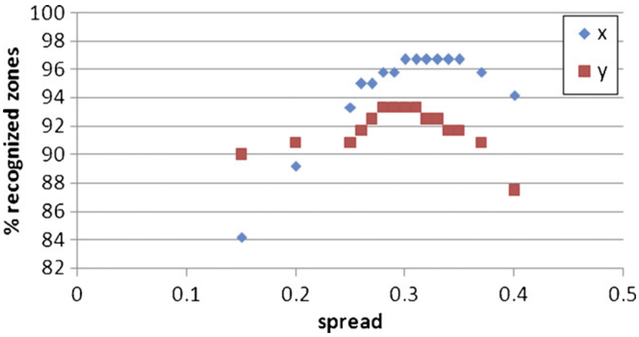

Fig. 13 – GRNN classification performance as a function of the spread parameter.

Table 3 – Performance of the neural networks for different distances from the camera

	Percentage of zone recognition zones			
	X (%)[a]	Y (%)[a]	X (%)[b]	Y (%)[b]
MLP	45	40	39	55
GRNN	100	100	80	60

[a] Far (53 cm).
[b] Near (41 cm).

for the y coordinate the best score (96.7%) is reached with a three-layer configuration. The GRNN has been tested for different values of the spread. As Fig. 13 shows, the optimum value of the spread has been set at 0.3 for the x and 0.33 for the y, leading to a performance of, respectively, 96.7% and 93.3% of zone recognition performance.

The nets above selected have been tested again, loosening some of the constraints to the head motion, in particular concerning the distance to the camera. The test set is in this second situation composed of two head positions, one far, at 53 cm from the screen and the other closer, at 41 cm from the camera. The different MLP configurations have shown a low capability of calculating the gaze correctly for these two distances, while the GRNN has shown a very good performance for the higher distance. In the case of close distance, some errors occur especially for the y coordinate estimation (Table 3).

According to the obtained results we can state that the MLP structure gives more accurate results for a given distance to the camera, while the GRNN is more robust than MLP even for changes in the z direction, still maintaining the percentage of correct recognition at a high level. For this reason we consider the GRNN structure more suitable for free head conditions.

In the following, an analysis of the global accuracy and robustness of the GRNN will be presented. The accuracy has been calculated in terms of mean and standard deviation of the gaze error, i.e. the error between the real observed position and the estimated values, expressed in terms of pixel (e_{pxl}) and angular degrees (e_{degree}), according to the following equation:

$$e_{\mathrm{degree}} = \arctan \frac{e_{\mathrm{pxl}}}{d_{\mathrm{pxl}}}$$

with d_{pxl} representing the distance between the subject and the screen plane expressed in pixels. As in the previous paragraph, the percentage of zone recognition will be used as a measure of the accuracy of the system.

Two kinds of results are reported, first considering each trial separately to determine the robustness to head movements (Table 4), then by reporting the accuracy on each zone over the different trials (Table 5 and Fig. 14) to highlight the distribution of the accuracy over the screen.

The estimated mean accuracy is approximately 1.6° on x direction and 2.6° on y direction with a standard deviation, respectively, 1.4° and 1.9°, leading to a rate of successful recognition of 94.7%. Exceptions have occurred in three zones, where the performance decreases to values under 90%.

4.3. *Performance comparison*

In this section we have compared the performance of the neural approach with a polynomial mapping function that has been often used in the context of gaze tracking based on geometric feature relations. In particular, Eq. (1) used in the infrared-based systems [13] has been implemented.

As mentioned in Section 2.1, the pupil-glint vector used in the infrared-based techniques is not reproducible in the view-based approaches, and thus a new vector has been considered as a proxy of the pupil-glint: it is obtained by connecting the pupil with the midpoint of the segment connecting the two corners.

Table 4 – Accuracy of the GRNN for nine different test trials

Trial	Distance (cm)	Mean value ± standard deviation (°, pixels)		Percentage of zone recognition	
		X	Y	X	Y
1	48	1.6 ± 1.2, 36 ± 26	3.3 ± 2.1, 78 ± 46	100	80
2	47	1.4 ± 0.9, 32 ± 21	1.2 ± 0.8, 31 ± 20	100	100
3	46	2.2 ± 1.3, 47 ± 27	4.1 ± 3.0, 91 ± 63	93.3	66.7
4	46,5	1.7 ± 1.2, 38 ± 27	1.3 ± 1.1, 32 ± 26	100	100
5	45	1.3 ± 1.1, 29 ± 24	1.5 ± 0.8, 37 ± 19	100	100
6	45	2.0 ± 1.8, 45 ± 40	3.5 ± 2.1, 80 ± 45	93.3	93.3
7	45	1.5 ± 1.0, 34 ± 25	1.8 ± 1.8, 43 ± 42	100	100
8	45	1.9 ± 1.3, 61 ± 28	2.4 ± 1.5, 59 ± 33	86.6	100
9	53	1.2 ± 1.3, 29 ± 30	2.8 ± 1.2, 66 ± 26	100	100
Mean		1.7 ± 1.2°	2.4 ± 1.6°	97	93.3

Table 5 – Accuracy of the GRNN on the 15 zones

Zone	Mean value ± standard deviation (°)		Zone recognition (%)
	X	Y	
1	1.8 ± 1.8	2.1 ± 1.3	93.3
2	0.7 ± 3.1	3.4 ± 2.2	93.3
3	1.0 ± 1.5	2.5 ± 1.0	100
4	1.4 ± 1.7	1.8 ± 1.3	100
5	1.7 ± 1.0	1.2 ± 1.6	100
6	2.8 ± 0.7	3.5 ± 3.1	93.3
7	0.2 ± 2.0	2.6 ± 3.1	93.3
8	0.4 ± 1.1	2.8 ± 2.4	93.3
9	0.3 ± 2.0	2.5 ± 2.7	93.3
10	4.2 ± 1.5	2.3 ± 2.8	80
11	3.9 ± 1.4	3.5 ± 1.8	100
12	0.3 ± 2.5	2.7 ± 1.8	100
13	0.3 ± 1.9	2.8 ± 2.3	100
14	0.1 ± 1.9	4.0 ± 2.5	93.3
15	2.7 ± 1.6	5.9 ± 3.1	86.7
Mean	1.4 ± 1.7	2.9 ± 2.2	94.7

The quadratic function takes as inputs the two components of the vector returning the coordinates of the screen point. A 15-point calibration and a least square solution have been used to solve the over-constrained problem, as proposed by Morimoto et al. [13]. The results depicted in Fig. 15 demonstrate the better performance of the neural approach as compared to the polynomial interpolation method in terms of mean values and standard deviations, showing an accuracy of 1.7 ± 1.2° for the x direction and 2.4 ± 1.6° for the y direction, while the accuracy of the quadratic function, respectively, of 2.9 ± 1.9° and 3.6 ± 1.9° for x and y directions.

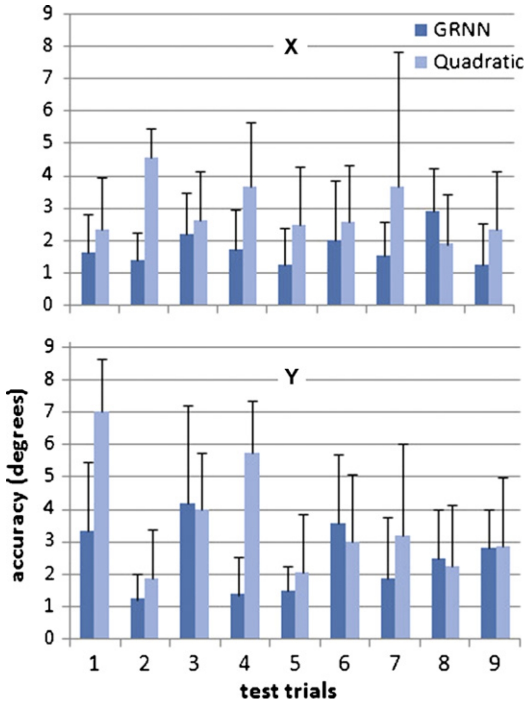

Fig. 15 – Comparison to the quadratic mapping.: mean values (histograms) and standard deviations of the error in degrees for the different trials.

5. Discussion

The proposed REGT system shows reliable and accurate global results. The uncertainty of gaze estimation has been proven to come from two main factors. The first one refers to the eye

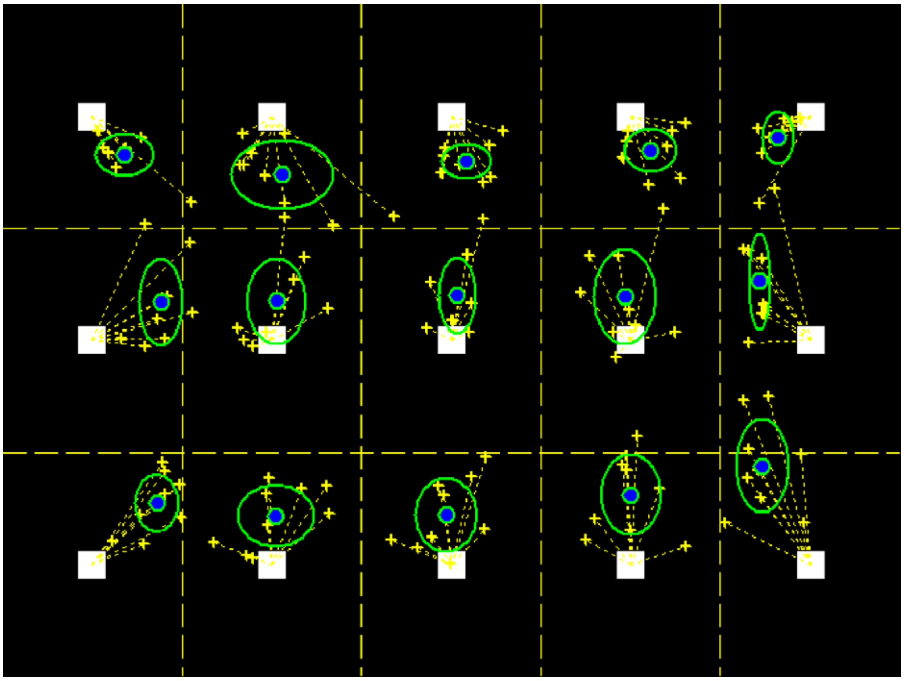

Fig. 14 – Accuracy evaluation. White squares represent the observed positions. The crosses correspond to the estimated values (1 for each test trial). The small circles represent mean values among the trials, while the ellipses stand for the standard deviations in x and y direction.

features tracking: in some extreme cases, if the gaze is directed towards the very lowest part of the screen and off the centre, the iris tracking algorithm does not achieve a high accuracy, due to occlusions from the eyelids and significant changes in the iris shape, so that the template matching is not as accurate as for the intermediate positions. This is what happens for the zones 14 and 15 (low right in the screen, see Fig. 14) where the estimation accuracy increases to higher, yet acceptable, values of error.

The second source of inaccuracy is due to the nature of the input set, which might describe a non-predictable distribution over the 12-dimensional space in presence of head movements. The neural mapping seems to represent a proper solution for this problem, overcoming the high non-linearity of head–eye movements. The GRNN structure has been proven to be more effective than MLP in compensating for head motion along the z direction, while the MLP network seems to achieve higher values of accuracy for distances very similar to the ones present in the training set. Since the system is aimed at compensating for movement in a 3D space, the GRNN has been considered the most appropriate solution. The neural mapping outperforms the quadratic mapping function, as one would expect by considering the possibility of having the user look at the same point on the screen by having different eye feature configurations with different head positions.

The system has also been proven to be robust to light changes. By using an initialization procedure based on the movements of the eyelids during blinking, the feature extraction and tracking is almost independent from the presence of light changes, both in intensity and direction. Moreover, this preliminary procedure attains to automatically initialize the procedure, avoiding an external user to select the features to track.

The estimated accuracy of the system is pretty good, yielding values comparable with most of view-based REGT ones, except for the work of Xu et al. [18], where the accuracy is, however, calculated through the leave-one-out procedure, whose use is controversial, since it usually underestimates the true prediction error [36].

As for the classification performance, results are in favour of a very high classification rate under natural head motion with a 15-zone user interface environment, which represents a valuable trade-off between usability and timing cost.

The five-trial calibration procedure, necessary to let the system learn and correctly estimate the gaze direction, is user specific and requires about 8 min (5 min of visual task plus 1 min of rest between two consecutive trials): the burden time is acceptable for user comfort.

6. Conclusions and future goals

The proposed remote eye gaze tracker is based on a novel approach in which eye feature extraction and tracking is combined with neural mapping, with the purpose of improving robustness, accuracy and usability under natural conditions. By working on visible light spectrum and by extracting features based on blink detection, the method has shown not to be affected by light changes.

The core of the proposed approach overcomes most of the issues induced by head movement, that is one of the most serious problems in REGT systems. Two neural network structures have been designed to learn the gaze direction from the user, under natural head motion: head shift was performed during the training sessions and a test session with head motion in 3D has been performed. The aim of the testing was to test the system under natural head movements during a visual task, maintaining a comfortable posture.

The obtained results (global average accuracy of around 2° and classification rate of 95% over 15 zones) confirm the reliability and robustness of the proposed system, in which the neural mapping has been proven to outperform traditional quadratic approaches.

The proposed system makes use of simple, low cost and commercially available hardware, since no further device than a webcam is needed.

Future efforts will be devoted to develop new strategies to address the issue of larger head movements. In this context a convincing solution can be represented by a combination of a decision tree approach with a neural network design, to explore and exploit the classification and approximation abilities of the neural systems in the direction of making remote eye gaze trackers effective solutions both in the assistive technology framework and in rehabilitation contexts.

Conflicts of interest

The authors allege that no financial or personal relationships with any other people and organisation have inappropriately influenced the work here submitted.

Acknowledgements

This research activity has been partially funded by the Italian Ministry of University and Research (MiUR). The help of the two anonymous reviewers is greatly acknowledged. A special thanks to Roberto Moschini and Saba Gervasio (and family) for their technical help and suggestions.

REFERENCES

[1] C.H. Morimoto, M.R.M. Mimica, Eye gaze tracking techniques for interactive applications, Computer Vision and Image Understanding 98 (2005) 4–24.

[2] A.T. Duchowski, A breadth-first survey of eye tracking applications, Behavior Research Methods, Instruments, and Computers 34 (2002) 455–470.

[3] D.W. Hansen, A.E.C. Pece, Eye tracking in the wild, Computer Vision and Image Understanding 98 (2005) 155–181.

[4] T. Hutchinson, K.J. White, K. Reichert, L. Frey, Human–computer interaction using eye-gaze input, IEEE Transactions on Systems, Man, and Cybernetics 19 (1989) 1527–1533.

[5] Tobii Technology AB, Stockholm, Sweden, http://www.tobii.se.

[6] SMI, SensoMotoric Instruments GmbH, Teltow, Germany, http://www.smi.de.

[7] Eye Response Technology, Charlottesville, USA, http://www.eyeresponse.com.

[8] D.A. Robinson, A method of measuring eye movement using a scleral search coil in a magnetic field, IEEE Transactions on Biomedical Engineering 10 (1963) 137–145.

[9] M.J. Coughlin, T.R. Cutmore, T.J. Hine, Automated eye tracking system calibration using artificial neural networks, Computer Methods and Programs in Biomedicine 76 (2004) 207–220.

[10] L.E. Hong, M.T. Avila, I. Wonodi, R.P. McMahon, G.K. Thaker, Reliability of a portable head-mounted eye tracking instrument for schizophrenia research, Behavioural Research Methods 37 (2005) 133–138.

[11] M. Adjouadi, A. Sesin, M. Ayala, M. Cabrerizo, in: K. Miesenberger, J. Klaus, W. Zagler, D. Burger (Eds.), Computers Helping People with Special Needs, Springer, Berlin/Heidelberg, Germany, 2004, pp. 761–769.

[12] S. Shih, J. Liu, A novel approach to 3d gaze tracking using stereo cameras, IEEE Transactions on Systems, Man, and Cybernetics 3 (2003) 1–12.

[13] C. Morimoto, D. Koons, A. Amir, M. Flickner, Pupil detection and tracking using multiple light sources, Image and Vision Computing 18 (2000) 331–336.

[14] Q. Ji, Z. Zhu, Non-intrusive eye and gaze tracking for natural human computer interaction, MMI-Interaktiv 6 (2003), ISSN 1439–7854.

[15] D.H. Yoo, M.J. Chung, A novel non-intrusive eye gaze estimation using cross-ratio under large head motion, Computer Vision and Image Understanding 98 (2005) 25–51.

[16] S. Baluja, D. Pomerleau, Non-intrusive gaze tracking using artificial neural networks, in: J.D. Cowan, G. Tesauro, J. Alspector (Eds.), Advances in Neural Information Processing Systems (NIPS), vol. 6, Morgan Kaufmann Publishers, San Francisco, CA, 1994, pp. 753–760.

[17] K.H. Tan, D.J.J. Kriegman, N. Ahuja, Appearance-based eye gaze estimation, in: Proceedings of the IEEE Workshop on Applications of Computer Vision-WACV02, 2002, pp. 191–195.

[18] L.Q. Xu, D. Machin, P. Sheppard, A novel approach to real-time non-intrusive gaze finding, in: Proceedings of the 9th British Computer Vision Conference-BMVC, 1998, pp. 428–437.

[19] Y. Ebisawa, Improved video-based eye-gaze detection method, IEEE Transactions On Instrumentation and Measurement 47 (1998) 948–955.

[20] Z. Zhu, Q. Ji, Eye gaze tracking under natural head movements, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 918–923.

[21] J. Zhu, J. Yang, Subpixel eye gaze tracking, in: Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 124–129.

[22] R.J.K. Jacob, K.S. Karn, Eye tracking in human-computer interaction and usability research: ready to deliver the promises, in: R. Radach, J. Hyona, H. Deubel (Eds.), The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research, North-Holland/Elsevier, Boston, MA, 2003, pp. 573–605.

[23] Z. Zhu, Q. Ji, Novel eye gaze tracking techniques under natural head movement, IEEE Transactions on Biomedical Engineering 99 (2007) 1–11.

[24] R. Newman, Y. Matsumoto, S. Rougeaux, A. Zelinsky, Real time stereo tracking for head pose and gaze estimation, in: Proceedings of the fourth IEEE international conference on automatic face and gesture recognition, 2000, pp. 122–128.

[25] D. Beymer, M. Flickner, Eye gaze tracking using an active stereo head, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. II, Madison, WI, 2003, pp. 451–458.

[26] T. Ohno, N. Mukawa, A Free-head, Simple Calibration, Gaze Tracking System That Enables Gaze-Based Interaction, ETRA 2004: Eye Tracking Research and Applications Symposium, 2004, pp. 115–122.

[27] K.R. Park, A real-time gaze position estimation method based on a 3D eye model, IEEE Transactions on Systems, Man and Cybernetics 37 (2007) 199–212.

[28] D. Torricelli, M. Goffredo, S. Conforto, M. Schmid, T. D'Alessio, A novel neural eye gaze tracker, in: Proceedings of the 2nd International Workshop on Biosignal Processing and Classification—Biosignals and Sensing for Human Computer Interface (BPC 2006), 2006, pp. 86–95.

[29] T. Kawaguchi, D. Hidaka, M. Rizon, Detection of eyes from human faces by Hough transform and separability filter, in: International Conference on Image Processing 2000 Proceedings, 2000, pp. 49–52.

[30] Parker, R. James, Algorithms for Image Processing and Computer Vision, John Wiley & Sons, Inc., New York, 1997, pp. 23–29.

[31] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (6) (1986) 679–698.

[32] W.K. Pratt, Digital Image Processing, Wiley, New York, 1978.

[33] M. Goffredo, M. Schmid, S. Conforto, T. D'Alessio, A markerless sub-pixel motion estimation technique to reconstruct kinematics and estimate the centre of mass in posturography, Medical Engineering & Physics 28 (2006) 719–726.

[34] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998.

[35] D.F. Specht, A general regression neural network, IEEE Transactions on Neural Networks 2 (1991) 568–576.

[36] H.A. Martens, P. Dardenne, Validation and verification of regression in small data sets, Chemometrics and Intelligent Laboratory Systems 44 (1998) 99–121.