

# Hybrid method based on topography for robust detection of iris center and eye corners

ARANTXA VILLANUEVA, VICTORIA PONZ, LAURA SESMA-SANCHEZ, MIKEL ARIZ, SONIA PORTA and RAFAEL CABEZA, Public University of Navarre

---

A multi-stage procedure to detect eye features is presented. Multiresolution and topographic classification are used to detect the iris center. The eye corner is calculated combining valley detection and eyelid curve extraction. The algorithm is tested in the BioID database and in a proprietary database containing more than 1200 images. The results show that the suggested algorithm is robust and accurate. Regarding the iris center our method obtains the best average behavior for the BioID database compared to other available algorithms. Additional contributions are that our algorithm functions in real time and does not require complex post processing stages.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation, e.g. HCI]: User Interfaces—*Input devices and strategies* ; I.4.9 [Image Processing and Computer Vision]: Applications—*Applications*; I.5.4 [Pattern Recognition]: Applications—*Computer Vision*

General Terms: Algorithms

Additional Key Words and Phrases: Iris detection, eye tracking, topography, corner detection

---

## 1. INTRODUCTION

Research on eye detection and tracking has attracted much attention in the last decades. Since it is one of the most stable and representative features of the subject, eye detection is used in a great variety of applications, such as subject identification, human computer interaction [Morimoto and Mimica 2005] and gesture recognition [Tian et al. 2000][Bailenson et al. 2008].

Gesture recognition based on analytic approaches relies in face features detection such as eyes and eyebrows among others [Mitra and Acharya 2007]. In the survey provided by Mitra and Acharya [2007] it is reported that face and gesture recognition have potential applications in areas such as criminal identification, surveillance, video document retrieval, telecommunication, high-definition television (HDTV), medicine and human computer interfaces. The work by Min et al. [2011] about recognition of faces demonstrates that, in agreement with the psychophysical findings, eye regions play the most important role in face recognition. Additional examples about the importance of eye detection are provided in [Bicego et al. 2006]. They use Scale Invariant Feature Transform (SIFT) features for face identification. One of the methods takes into account the fact that most part of the face information is located around the eyes and the mouth. Thus, the position of these landmarks is required for the anal-

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 2010 ACM 1544-3558/2010/05-ART1 \$10.00

DOI 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

ysis. Taheri et al. [2011] reinforce in their paper the importance of facial gestures recognition based on landmarks placed in the face area, including eyes. We address the reader to these references to illustrate the importance of robust and accurate eye detection methods.

However, human computer interaction based on eye information is one of the most challenging research topics in the recent years. According to the literature, the first attempts to track the human gaze using cameras began in 1974 [Merchant et al. 1974]. Since then, and especially in the last decades, much effort has been devoted to improving the performance of eye tracking systems. Although no brand names will be provided, the currently available commercial systems are a reference for the feasibility of this technology. However, the applications of such systems are limited, consisting mainly in gaze movement analysis and human-computer interaction for severely disabled [Majaranta et al. 2011]. Human gaze can be considered an indicator of subject's interest and attention; hence, eye tracking systems have diverse applications in numerous fields such as psychology and market research. The initial motivation for the research in the eye tracking field was mainly to explore the behavior of human eye and the connection with brain cognitive processes [Monty et al. 1976].

Regarding human-computer interaction, the availability of high performance eye tracking systems has provided, in the past decades, advances in fields such as usability research [Ellis et al. 1998] [Poole and Ball 2005] and interaction for severely disabled people [Bolt 1982] [Starker and Bolt 1990] [Vertegaal 1999]. Gaze tracking systems can be used to determine the fixation point of an individual on a computer screen, which can in turn be used as a pointer for interaction with the computer. Thus, severely disabled people who cannot communicate with their environment using alternative interaction tools can perform several tasks by means of their gaze and a tracker. Performance limitations, such as head movement constraints, limit the employment of the gaze trackers as interaction tools in other areas. Moreover, the limited market for eye tracking systems and the specialised hardware they employ, increase their prices. The eye tracking community has identified new application fields, such as video games or the automotive industry, as potential markets for the technology. However, simpler (i.e., lower cost) hardware is needed to reach these areas.

In the recent years, research into low-cost eye tracking systems based on web cams has been identified as a promising (and necessary) endeavour for scientists in the field. Web cam based eye tracking creates new image processing obstacles that need to be overcome. Although web cams offer acceptable resolutions for eye tracking purposes, the optics used provide a wider field of view in which the whole face appears. By contrast, most of the existing high-performance eye tracking systems employ infrared illumination. Infrared light-emitting diodes provide a higher image quality and produce bright pixels in the image from infrared light reflections on the cornea named as glints. Although some works suggest the combination of light sources and web cams to track the eyes [Sigut and Sidha 2011], the challenge of low-cost systems is to avoid the use of light sources to keep the systems as simple as possible; hence, the image quality decreases. In Figure 1, the images obtained by a high performance eye tracker and a web cam are compared. High-performance eye tracking systems usually combine glints and pupil information to compute the gaze position on the screen. Accurate pupil detection is not feasible in web cam images, and most works on this topic focus on iris center. In order to improve accuracy, other elements such as eye corners or head position are necessary for gaze estimation applications, apart from the estimation of both irises. In the work by Ince and Yang [2009], they consider that the horizontal and vertical deviation of eye movements through eyeball size is directly proportional to the deviation of cursor movements in a certain screen size and resolution. Fukuda et al. [2010] employ iris information and eyeball geometry information in their gaze estimation method. Other approaches use preprocessed eye regions to train a neural network as made by Sewell and Komogortsev [2010]. If user movement tolerance is required, as well as iris position, head position is needed. Using eye corners is a straightforward method to overcome this problem, and the corners are employed in several works

to improve gaze estimation accuracy as it is shown in Valenti et al. [2009]. Zhu and Yang [2002] work presents a webcam based eye tracking system. Although it uses infrared for image processing purposes, the relevant aspect of their paper is that they use iris and corner information for gaze estimation.



Fig. 1. Images of similar resolution obtained by a high performance eye tracking system (left) in which the optics used and two infrared light-emitting sources permit to obtain a focused image of the eye area. The image obtained by the web cam (right) captures a wider area of the scene.

Different approaches for detecting iris centers have been presented in recent years. Relevant reviews of the existing methods are discussed in Hansen and Ji [2010] and Timm and Barth [2011]. Different criteria can be used to classify existing methods for iris center detection. According to Timm and Barth [2011] these methods can be classified as (i) feature-based methods, (ii) model-based methods, and (iii) hybrid methods. Wang et al. [2007] classify these methods into holistic and abstractive methods. Several methods propose additional processing stages, such as Kalman filter or Mean Shift techniques to improve accuracy.

Recently, significant results have been achieved using iris curvature. Valenti and Gevers [2008] base their method on a characterisation of the eye using radially symmetrical patterns; thus, they use isophotes (curves connecting points of equal brightness) to detect the iris curvature. Accordingly, they propose a voting algorithm that is key to determine the iris center. Timm and Barth [2011] propose an objective function based on gradients to determine the iris center. The objective function is maximised for centers of circular regions. Curvature-based methods can fail when the eye edges are not completely visible (extreme eye rotation, eyelid occlusion, etc.) or when eyebrows are prominent. In those cases, other curved elements in the eye region, such as eyebrows or wrinkles, can represent the contour of the most-voted center.

Among the non-curvature-based methods, algorithms based on topographical characteristics try to label each pixel according to grey level changes in the pixel neighborhood [Ponz et al. 2011]. The patterns of these topographic labels capture information about the original three-dimensional object in the scene and about the illumination [Pong et al. 1985]. According to the nomenclature employed in topography-related works, iris regions can be labeled “pits”. Wang and Yin [Wang et al. 2007] propose the construction of the topographic manifold of the image and based on Bhattacharyya kernel, a support vector machine (SVM) is then applied to select the proper iris pair from the pit-labeled candidates. A similar technique is employed in Ferdowsi and Ahmadyfard [Ferdowsi and Ahmadyfard 2008]; once the pixels are labeled using topographical criteria, regional-invariant moments of the topographic image are employed to detect the eyes according to a Bayesian classifier.

Regarding eye corner estimation, we find works in which corners are detected as a result of facial features detection methods. Recently, Dibeklioglu et al. [2011] and Belhumeur et al. [2011] have presented relevant works in the area; however, both of them require training sets to detect facial features. In the same manner, works in which a specific detection of the eye corner is carried out have been presented lately. In Zhu and Yang [2002], they present a method based on spatial filtering and corner shape masks to detect eye corners. Zhou et al. [2011] use Harris detector and texture analysis to determine eye corners in the image. Haiying and Guoping [2009] apply weighted variance projection function to determine a rough corner area and Harris corner detector to improve the accuracy. Harris

detector is also employed in Xu et al. [2008] to detect candidate points followed by a postprocessing to determine eye corners. The main problem of the methods based on general purpose corner detectors is the fact that eye corners can largely vary between subjects. Corner detectors such as Harris and Susan can fail since eye corners do not always present the required image characteristics.

In this paper, we present a novel feature-based method that uses topography and curve extraction to detect iris center and outer eye corner position. Topography techniques are used for both, iris center and outer eye corner detection. Multiresolution images are employed to provide robustness to the iris center algorithm, while curve extraction supports outer eye corner estimation. To keep the algorithm as simple as possible, learning or training stages are avoided. The proposed solution is simple, and the results obtained are comparable to other methods that use complex postprocessing stages. The proposed algorithm was tested on two databases. BioID represents a challenging database used as test benchmark in many works [Dibeklioglu et al. 2011] [Belhumeur et al. 2011] [Timm and Barth 2011]. The dataset consists of 1521 gray level images with a resolution of 384x286 pixels. Each image shows the frontal view of a face of one out of 23 different test persons. BioID also provides text files with labels for eye features. The method is also tested on a proprietary database. It contains higher resolution images, i.e. 800x600 pixels, of more than 100 subjects gazing at different points in the screen. Each subject gazes at 12 points uniformly distributed in the screen. The database images have been labeled by three observers in order to make the labels more independent from the observation. The final labels are calculated as the average of the three observations.

In Section 2, the proposed algorithm for detecting the iris center is presented. Then, corner detection is described in Section 3. An evaluation of the methods is performed in Section 4. Finally, the conclusions are presented in Section 5.

## 2. MULTIREOLUTION-BASED IRIS DETECTION METHOD

A novel method for detecting the iris center is proposed that is based on topography and uses multiresolution analysis. The idea of using topography for iris center detection has been previously employed by Wang et al. [Wang et al. 2007], but their approach is computationally more expensive and exhibits lower accuracy than ours, as it will be demonstrated later. We incorporate a multi-stage method that is normally employed in feature-based methods. A face detector is used to detect the face area [Viola and Jones 2004]. The Viola Jones detector is widely used and its performance has been demonstrated. Once the face is detected, rough eye regions are determined (see Figure 2). Our method is then performed in these regions. The iris center detection method is described in the following paragraphs.



Fig. 2. The detection scheme: the face is first detected, and then the eye regions are roughly estimated. The iris center detection algorithm is applied to each eye image.

From image topography theory perspective, image pixels can be labeled according to their grey level and the intensity of their neighbouring pixels [Wang et al. 2007]. Given the image  $f(x, y)$ , the labeling

process is performed using the Hessian matrix eigenvalues and the gradient vector behavior. Given a pixel at position  $(x, y)$  the Hessian matrix is calculated as:

$$\mathbf{H}(x, y) = \begin{pmatrix} \frac{\partial^2 f(x, y)}{\partial x^2} & \frac{\partial^2 f(x, y)}{\partial x \partial y} \\ \frac{\partial^2 f(x, y)}{\partial x \partial y} & \frac{\partial^2 f(x, y)}{\partial y^2} \end{pmatrix} \quad (1)$$

From the eigenvalue decomposition of  $\mathbf{H}$ ,  $\lambda_1$  and  $\lambda_2$  eigenvalues are obtained. Differentiation filters based on Chebyshev polynomials are used to approximate topographic labels computation defined for continuous functions to discrete signals [Meer and Weiss 1992].

Image topography allows the labeling of pixels as ridge and pits among others. Thus, the center of the iris can be considered to be a valley since, ideally, intensity increases in all directions. In topography, these points are called a “pit”. A pixel is classified as a “pit” if the following conditions are satisfied:

$$\|\nabla f(x, y)\| = 0, \lambda_1 > 0, \lambda_2 > 0 \quad (2)$$

As described in the image topography literature, the gradient is zero when it can be approximated by zero, and the eigenvalues are greater than zero when their values are significantly large. Hence, threshold values are used to determine whether a pixel can be described as a pit.

As a preprocessing step, a Gaussian low-pass filter is applied to the image before the topography analysis is performed. Moreover, a morphological opening is used to remove possible glints inside the iris that can affect the labeling procedure. Iris centers are not the only points that can be characterized as valleys in the eye area. Eye corners and eyebrow parts are frequently classified as pits, as shown in Figure 3. Wang et al. [2007] solve the indetermination of multiple candidates by applying a SVM (Support Vector Machine) scheme that is trained as it will be later shown in Section 4. We suggest a faster and more accurate detector that does not rely on time consuming learning procedures. To avoid the training stage, we propose (i) removing the eyebrows in a prior stage that is based on the integral projection method and (ii) performing a multiresolution analysis that is followed by the processing of the valleys at different resolution levels to determine the correct iris center.



Fig. 3. Eyebrow pixels and eye corners are frequently detected as pits, as they present valley-shaped intensities.

## 2.1 Eyebrows removal

In the eye, apart from the iris, prominent eyebrow parts are normally detected as pits. A simple method is proposed for rejecting eyebrow pixels as candidates. An integral trapezoidal projection is applied to the eye region to detect the eyebrows, the horizontal integral projection. An eyebrow is detected if a maximum is found between two minimums in the first 50% of the eye region rows. A point is defined as a maximum if its value exceeds the mean value of the integral projection, while values below half of the mean value are required for a point to be classified as a minimum (see Figure 4).

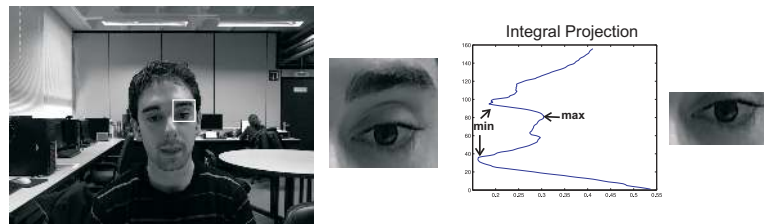


Fig. 4. Once the eye region is selected, the integral projection is used to find the eyebrow. An eyebrow is identified when a maximum is detected between two minimums in the integral projection.

Once the maximum is detected, the image is cropped by eliminating the rows ranging from the first row to the position of the maximum of the integral projection. When the eyebrow is not dark enough, that maximum is not found. In these cases, the eyebrow is not removed and a negative flag is assigned, meaning that eyebrows are not dark enough to be classified as a pit in the topography analysis and won't cause the algorithm to fail.

## 2.2 Valley detection at different resolutions

The idea of multiresolution image analysis has been largely used in many applications [Dyer 1987] [Bertolino and Montanvert 1996]. Processing the image at different scales adds a new dimension to the problem. Images at lower resolutions represent the original image but at different sizes. At lower resolutions the ability to represent details is reduced. The multiresolution theory was introduced by Lindeberg [1994] where it was used for detail removal. Multiresolution versions of the image simplify the contents and allow to search for objects with known model but unknown size. In the case under study, both iris and eye corners behave as valleys, i.e. same model, but with different sizes. Regardless of the resolution level used, the same filters, i.e. kernels, are applied to detect pits. The procedure is as follows:

- A gaussian low pass filter is applied to reduce noise. The size of the filter is varied according to the dimensions of the detected face; this compensates for size differences due to the placement of the subject at different distances from the camera. However, once the filter is designed, the same filter is applied for every scale of the image. For a user at a standard distance, i.e. 60 cm, a 15x15 kernel is used with  $\sigma = size_{kernel}/6$ .
- Differentiation filters, mentioned in the beginning of this section with 9x9 size, are applied to calculate Hessian matrix in every resolution. It is straightforward to deduce that valleys of different sizes will behave differently at different resolutions. As the resolution decreases, the lack of detail will distort valley shaped areas.

In order to find out which resolutions are most appropriate, relative sizes of iris and eye corner valleys have been analyzed in different databases. Although there is not a fixed proportion, we can assume a 2:1 relation between iris and eye corner sizes. Hence, reducing image resolution by half would remove eye corners, and comparing both resolutions would be enough to differentiate between both features. However, this value can slightly change between subjects; furthermore, this relationship can vary for the same user when gazing at different points on the screen. Thus, both features can disappear at the lower resolution. In order to provide more robustness to the method, an intermediate resolution at 0.75 is introduced.

Therefore, topography-based pits detection is performed at maximum resolution, 0.75 of the maximum resolution and 0.5 of the maximum resolution. For each resolution, different pit regions are obtained according to the procedure previously described. The coordinates of the candidates obtained

at lower resolutions are re-mapped to the maximum resolution. When the pixels of the pit region are re-mapped, the connectivity of those regions is lost. To recover that connectivity, a dilation is performed to fill the gaps that are created between the pixels of the pit region.

As expected, the number of valleys decreases as the image resolution is reduced. As mentioned before, the underlying assumption is that the iris center is the most stable valley at different resolutions, while other possible valleys, such as eye corners, vary with different resolutions. An example is shown in Figure 4. The original eye image is shown together with the valleys that are calculated at different resolutions. The pixels that are detected as pits in the three resolutions are considered to be the center of the iris. In practice, however, there is not always complete overlapping of the valley pixels, when the three resolutions are compared; hence, additional processing is performed for these images.



Fig. 5. Images of the right eye in which the pits obtained at different resolution levels are mapped. Left, maximum resolution, center 0.75 and right 0.5 resolution.

The centroids of the valley regions at the maximum resolution are taken as the reference  $C_{i0}, i = 1..n0$ , where  $n0$  is the number of centroids. The distances of these centroids and the centroids of the valleys at lower resolutions are computed, i.e.  $\{C_{j1}, j : 1..n1\}$  and  $\{C_{k2}, k : 1..n2\}$ , where  $n1$  and  $n2$  denote the number of centroids at the 0.75 and 0.5 resolutions respectively. For each one of the centroids,  $C_{i0}$  the minimum distance to  $\{C_{j1}, j : 1..n1\}$  is calculated  $D01_i, i = 1..n0$  where,

$$D01_i = \arg_j \min\{norm(C_{i0} - C_{j1})\} \quad (3)$$

In the same manner,  $\{D02_i, i = 1..n0\}$  is calculated as the subset of the minimum distances between the centroids at the maximum resolution and the centroids at 0.5 of that resolution:

$$D02_i = \arg_k \min\{norm(C_{i0} - C_{k2})\} \quad (4)$$

The center for which the average distance to a center at any of the other two resolutions is minimal is selected as iris center. In this manner, the algorithm selects the most stable center across resolutions but which can slightly move between resolutions. The point detected as the iris center is the centroid at the maximum resolution for which the average distance to the lower resolutions is minimum:

$$C_{iris} = \arg_i \min\left\{\frac{D01_i + D02_i}{2}\right\} \quad (5)$$

### 3. OUTER EYE CORNER DETECTION

After the center of the iris has been detected, outer eye corner detection is launched. A rectangular mask is applied to find the outer eye corner area. Depending on the eye, i.e. left or right, the mask will be placed to the left or right of the iris center. The size of the mask is proportional to the distance between the two iris centers  $d_i$ , i.e.  $[width = 0.4d_i, height = 0.2d_i]$  (see Figure 6a and 6c). The horizontal coordinate of the mask is the iris center while, vertically, the mask is located  $0.07d_i$  higher than the iris center. As mentioned before, from image topography perspective, eye corners are also considered as valleys. A straightforward approach would be to calculate the valleys located in the outer area of the

iris center. Although as first approach this could be valid and acceptable for some of the subjects, outer corners valleys are not as stable as the iris center valley. Depending on the subject, lower thresholds are needed to label a pixel as a pit, which produces additional valleys close to the outer eye corner. These additional valleys are located mainly in the eyelids and parts of the iris present in the selected processing window. Hence, the method used to detect the iris center is not feasible to detect the outer corner of the eye. Eye corner characterization, i.e modeling is not straightforward due to its elastic properties [Moriyama et al. 2006]. In this work, we suggest to detect the upper eyelid to strengthen the outer eye corner detection. To this end, an eyelid detector is implemented based on an edge detection and a curve extractor algorithm.

### 3.1 Eyelid detection

Canny edge detector is applied to the mask selected. Eyelids horizontal edge component is stronger than the vertical component. Consequently, Canny edge detector is modified and only the horizontal component is taken into account. The obtained edges are connected components of pixels located in the boundary between two regions in the image but not necessarily complete curves. Hence, a post processing is needed to link edge pieces into complete curves of the image.

End points for every edge calculated by the Canny detector are analyzed. For each end point, gaps are filled if this end point is nearly connected (within 1 pixel distance) to another end point or edge segment. Once this procedure is finished, the curves in the eye region are obtained (see Figure 6d). Except for some cases in which glasses present many and strong reflections that occlude eye area, the eyelid extraction method has demonstrated to be feasible and robust. In order to select the eyelid among the existing connected components, the longest curve in the upper part of the iris center with initial point closer to the nose is selected.

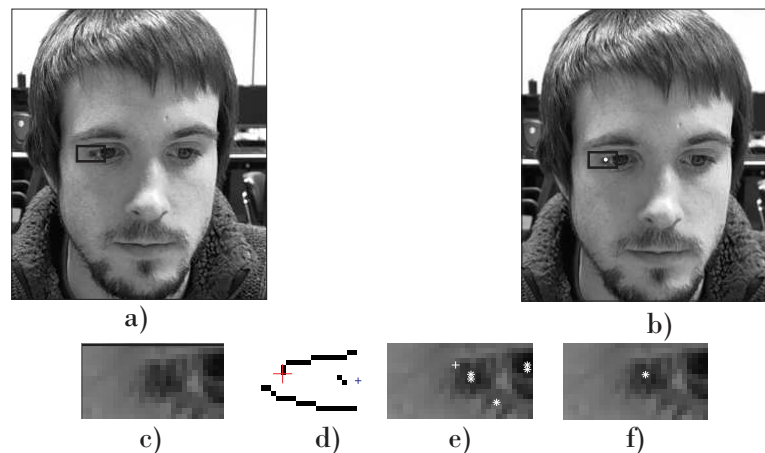


Fig. 6. a) Once the iris center is determined, a processing window is selected to estimate the eye corner. The size of the window is proportional to the distance between the two iris centers. b) The processing window and the estimated corner are shown. c) The processing window containing the eye corner. d) Eyelids curves. The bigger cross represents the estimated corner as the end point of the curve representing the eyelid while the smaller one represents the iris center. e) Valleys detected in the processing window after the topography analysis is done (shown with asterisks). The cross represents the point calculated as a consequence of eyelid curve detection. f) The pit closer to the end point of the eyelid is selected as outer eye corner.



Once the eyelid contour candidate is selected, the end point of the curve is determined as eye corner candidate, i.e. the end point is considered to be the farthest point with respect to the nose (see Figure 6d).

### 3.2 Eye corner accurate detection

One of the advantages of the eyelid detection method proposed is its robustness. However, the lack of accuracy can be considered as its main drawback. Eyelashes and eye wrinkles among others can slightly misplace the point detected. In other words, the method is robust since it can be assured that the corner candidate is located close to the eye corner but it cannot confirm that the obtained point is accurate enough.

As mentioned before, the eye corner can be classified as a pit under the topography rules. In contrast to eyelid detection, the eye corner pit detection is highly accurate only if the correct thresholds are selected, hence its robustness is low. If the thresholds selected are low enough the eye corner is detected together with other points in the eye area as shown in Figure 6e. The method proposed is to combine curve detection and topography in a robust and accurate eye corner detection algorithm.

The pit detection thresholds are therefore decreased until a sufficient number of pits are detected ( $> 6$ ). Once the pits are detected, the closest point to the corner candidate (extracted from the eyelid detection stage) is labeled as eye corner (see Figure 6f and Figure 6b).

## 4. EVALUATION

One of the most challenging collection of faces used to evaluate many iris detection algorithms is the BioID [Research 2001] database. The images are labeled, thus, together with the images, binary files containing iris centers and eye corners positions are provided. As mentioned before, the dataset consists of 1521 gray level images with a resolution of 384x286 pixels. BioID is the reference framework with which most of the methods in this field are compared.

GI4E (Gaze Interaction for Everybody) database has been created at the Public University of Navarra and is publicly available. Contacting the authors is required to access the database. The goal of this database is to simulate users interacting with a computer using their eyes, since this is oriented to be used by researchers in the field of gaze tracking. As mentioned in the introduction, it contains higher resolution images, i.e. 800x600 pixels, of more than 100 subjects gazing at different points in the screen, resulting in more than 1200 images. Labels of relevant points are also provided. Although the BioID database is a challenging database and a valid benchmark, we also think that GI4E images are closer to the ones that can be acquired by a webcam today.

### 4.1 Iris center

The error is calculated as the Euclidean distance between real (labeled) and calculated iris centers. Error calculus is performed separately for each one of the eyes and normalised with respect to the distance between the two iris centers. Among the possible methods to quantify the accuracy of the iris detection methods, the usual approach is to plot the percentage of the database images below a particular error value vs. that error value. According to the values that have been reported [Timm and Barth 2011], an error value  $< 0.25$  means that the estimated center is incorrectly located in the sclera, an error  $< 0.10$  represents an estimated center located within the area of the iris, and error values below 0.05 are within the diameter of the pupil. The accuracy obtained by the method is critical depending on the application. Detecting the iris area, i.e. error  $\approx 0.20$ , can be enough for face recognition or expression detection algorithms, but is clearly not acceptable for a gaze tracking system in which an error of 0.20 (3 pixels of error in the iris detection in standard working conditions for GI4E dataset) means 5 degrees of error in the visual angle. Accuracies of 1 degree are considered acceptable for high

performance eye trackers, while accuracies of 2-3 degrees have been reported for low cost eye trackers in the literature [Ince and Kim 2011].

The error is calculated for both eyes in each of the images, i.e.,  $e_{left}$  and  $e_{right}$ , and different accuracy curves are calculated for the maximum error, i.e.,  $e_{max} = \max(e_{left}, e_{right})$ , the mean error, i.e.,  $e_{avg} = (e_{left} + e_{right})/2$  and the minimum error, i.e.,  $e_{min} = \min(e_{left}, e_{right})$ .

As mentioned before, the preprocessing of the images consists of a Gaussian low-pass filtering. This filter was resized according to image size, and the area of the mask is approximated by one tenth of the image area. A morphological opening using a size 3 square structuring element is applied. In the case of the BioID database, we obtained slightly better results by adding image equalisation in the preprocessing stage. Images, for which the face detector failed or for which the eye regions were not correctly calculated, were cropped manually to avoid any influence of the eye region detection stage on the proposed algorithm. To analyse the 100% of the images in both databases, this cropping was applied to an 11% and 10% of the images of BioID and GI4E databases, respectively.

The Hessian matrix is calculated using filters defined in [Meer and Weiss 1992] of size 9x9. Moreover, the thresholds used as the zero reference are calculated for each image as  $0.5(\max(\|\nabla f\|) + \min(\|\nabla f\|))/2$  for the gradient and as 30% of the maximum values of  $\lambda_1$  and  $\lambda_2$  for each one of the eigenvalues. Based on our tests, the size of the filter and the thresholds selected do not produce significant changes in our results.

The same parameter values have been tested using images at higher resolutions from the GI4E database and for BioID database providing that the Gaussian filter size is modified according to the image size.

The following images show some of the results obtained for the BioID and GI4E databases. Correct detections for selected images are shown in Figures 7 and 8. The algorithm produces errors mostly in images that contain subjects with semi closed eyes or that wear glasses, as shown in Figures 9 and 10. Glasses often produce reflections from light sources and, when these reflections occlude the iris, the algorithm is not able to classify it as a pit. Similarly, for subjects that show semi closed eyes, grey levels around the iris center do not behave as a valley, i.e. pit, thus leading to a wrong detection. In Figure 11 the accuracy obtained by our algorithm in terms of  $e_{max}$ ,  $e_{min}$  and  $e_{avg}$  for the BioID database is shown.

The work by Timm and Barth [2011] presents an exhaustive review of most relevant iris center detection methods applied on the BioID database. Table I is based on the one reported by the authors; it shows the ranking of the alternative algorithms, including ours, for different error values for the BioID dataset. According to this table, our algorithm performs better on average. If we consider  $e_{max} \leq 0.05$  as the accuracy standard, however, is the third best method. [Valenti and Gevers 2008] report slightly (2%) better results at 0.05, however it relies on time consuming algorithms such as mean-shift clustering, SIFT features and a classifier as a postprocessing step. Furthermore, from our tests the procedure proposed by [Valenti and Gevers 2008] has demonstrated to be sensitive to the configuration of the algorithm and the parameters the method is based on. We would like to point out that the results obtained for our algorithm were calculated using all of the images in the BioID, (i.e., none of the images were eliminated).

A more exhaustive comparison has been performed between our method and two other iris detection algorithms. Since our algorithm does not rely on a postprocessing stage, the gradients based algorithm proposed by [Timm and Barth 2011] is considered to be an appropriate framework for comparison. In addition, both methods (i.e. our method and [Timm and Barth 2011]) are rather insensitive to changes in the design parameters of the algorithms. The method proposed by [Wang et al. 2007] has been also implemented and tested. Since [Wang et al. 2007] and our method share the same seminal idea and are both based on topography analysis, we consider it appropriate to be included in the comparison.



Fig. 7. Correct detections, for BioID (upper part) and GI4E (lower part) databases with the iris centers marked with white crosses.



Fig. 8. Enlarged eye areas of the images shown in Figure 7.

Timm and Barth [2011] propose an objective function based on image gradients. Basically, they select as iris center the point where most gradient vectors intersect. As is has been previously mentioned, our method and Timm and Barth [2011] perform a multi stage procedure, i.e. the algorithms are applied in a previously selected eye area based on a face detection stage. In order to carry out a fair comparison both methods have been applied to the same eye regions.

Wang et al. [2007] use image topography to label each pixel. Those pixels labeled as pits are considered to be candidates for iris center. In order to select the correct iris centers, a SVM is trained using image patches containing topography labels with positive and negative examples. These patches are calculated considering pairs of pits. The size and shape of the patches is a function of the line connecting both pits and the distance between them. Based on Bhattacharyya kernel, a SVM is applied.

Figure 12 compares the accuracies of the gradient-based algorithm [Timm and Barth 2011], the SVM algorithm Wang et al. [2007] and our method for the worse eye case in the BioID dataset. The graphs shown are the best results achieved with each one of the algorithms. The algorithm by Timm and Barth [2011] and our method are not affected by the parameters of the algorithm, whereas the algorithm by Wang et al. [2007] shows slight variations in the results when the training set is varied.



Fig. 9. False detections, for BioID (upper part) and GI4E (lower part), with the iris centers marked with white crosses.



Fig. 10. Enlarged eye areas of the images shown in Figure 9.

Table I. Algorithm ranking according to the minimum error value.

METHOD	$e_{max} \leq 0.05$	$e_{max} \leq 0.1$	$e_{max} \leq 0.15$	$e_{max} \leq 0.20$	$e_{max} \leq 0.25$	avg. rank
[Asadifard and Shanbezadeh 2010]	10	8	9	8	11	9.2
[Kroon et al. 2008]	6	7	-	-	3	5.3
[Valenti and Gevers 2008]	4	10	11	7	9	8.2
[Valenti and Gevers 2008]	1	5	7	4	4	4.2
[Türkan et al. 2007]	14	15	6	2	2	7.8
[Campadelli and Lanzarotti 2006]	7	9	10	9	10	9.0
[Niu et al. 2006]	5	4	3	5	8	5.0
[Chen et al. 2006]	-	6	-	-	12	9.0
[Asteriadis et al. 2006]	11	11	8	6	6	8.4
[Hamouz et al. 2005]	8	13	13	10	15	11.8
[Zhou and Geng 2004]	-	-	-	-	13	13.0
[Cristinacce et al. 2004]	9	1	2	4	7	4.6
[Behnke 2002]	13	7	5	3	5	6.6
[Jesorsky et al. 2001]	12	12	12	11	14	12.2
[Timm and Barth 2011]	2	3	4	5	5	3.8
<b>our method</b> [Ponz et al. 2011]	<b>3</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1.6</b>

In order to reject the effect of the training stage, the results shown in the graph for Wang et al. [2007] are those results that would be obtained if a perfect classification was possible, i.e. the pit selected as iris center is the closest to the real one. According to the graph, the method proposed by Timm and Barth performs slightly better at the critical point for  $e_{max} \leq 0.05$  (0.05% better). However, our method has better average behavior. The algorithm by Wang et al. [2007] presents a much lower performance for errors  $< 0.15$ , having a similar behavior for higher error values (always below our method). Remember that the results shown for Wang et al. [2007] are the ones assuming an accuracy of 100% in the classification, hence, worse results would be expected when SVM is introduced.

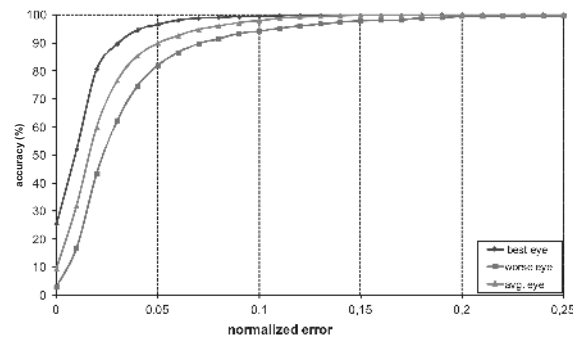


Fig. 11. The accuracy of the proposed method for the worse eye ( $e_{min}$ ), best eye ( $e_{max}$ ), and average eye ( $e_{avg}$ )

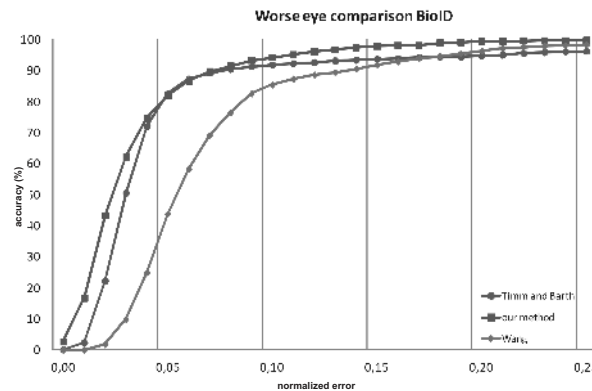


Fig. 12. A comparison of our method with those of Timm and Barth and Wang et al. for the worse eye case (BioID).

For the GI4E database, we compare worse eye performance of our method, Wang et al. [2007] and Timm and Barth [2011] method. The graph shown in Figure 13 represents the results. Compared to BioID database, all methods present higher performances. This is due to the fact that GI4E database presents images with a better quality in terms of resolution. At the critical point of 0.05, our method presents an accuracy of 93.92%, while the method by Timm and Barth [2011] obtains a slightly lower performance of 92.36%. The accuracy obtained by the method by Wang et al. [2007] is not comparable to the other two, being below 60%.

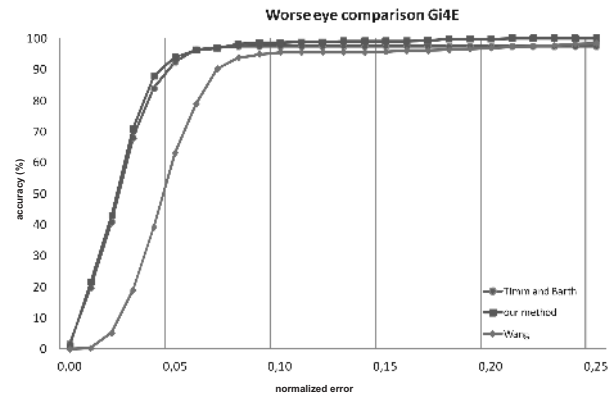


Fig. 13. A comparison of our method with those of Timm and Barth and Wang et al. for the worse eye case (GI4E).

In most of the cases, the method proposed by Wang et al. [2007] is able to determine the region of the image in which the iris center is contained, but with lower accuracy compared to our method. Furthermore, when the iris center is not detected as a pit in the image, the classifier does not consider it as a candidate. In Figure 14 the dot shows the point detected as a pit by the method proposed by Wang et al. [2007]. The iris candidate is wrongly detected in the eye corner. Hence, the classifier is not able to compensate for the error. The multi resolution approach used by our method permits to detect the pit corresponding to the iris center in lower resolutions, thus finding the correct center of the iris (the point is shown with a cross in Figure 14). In the original paper by Wang et al. [2007] results for two databases are provided: the Japanese female facial expression (JAFFE) database [Shih et al. 2008] and the facial recognition technology (FERET) database [Phillips 1998]. Since none of these databases were designed for gaze tracking purposes we have labeled them in order to mark iris centers more accurately. In the case of the FERET database the normalized error of the original marks with respect to our more accurate labels was 0.029 (the new labels are publicly available). If a perfect classifier is assumed, at the critical point of 0.05 the results obtained by the method proposed by Wang et al. [2007] are 92.5% and 94.5% for JAFFE and FERET databases, respectively, whereas our method obtains 91.55% and 96.94% for the same data sets. As mentioned before, the method by Wang et al. [2007] presents an acceptable accuracy for high quality images in which subjects gaze at the front, however, its performance decreases significantly when lower quality databases such as BioID are tested or when images with different gaze directions, e.g. GI4E database images, are employed as it has been shown in Figures 12 and 13.



Fig. 14. Since the iris center is not detected as a pit in the highest resolution, the method by Wang et al. does not consider it as input for the classifier (dot). However, our approach is able to calculate a better estimation of the iris center (cross).

Computational load has also been compared between Timm and Barth [2011] and our method. The method by Wang et al. [2007] has not been included in the comparison since it requires a previous training stage that makes it much more computationally expensive. The proposed algorithm has demon-

strated to work in real time and is about 16 times faster than the algorithm proposed by Timm and Barth [2011] in a standard situation. Figure 15(left) shows a plot of the processing time as a function of the number of pixels in the eye area for both algorithms for images of 800x600 resolution. As the figure shows Timm and Barth [2011], processing time increases more rapidly. Furthermore, the processing time of our algorithm increases more linearly as the size of the image increases (subject closer to the camera) while processing time of Timm and Barth [2011] rises up to approximately 20 sec per image for situations in which the user is placed close to the camera ( $\sim 25$  cm). In this configuration, our algorithm remains under 0.7 seconds per image. This algorithm, i.e. [Timm and Barth 2011], is based on a voting procedure in which all the pixels of the eye region are voted by the pixels for which the gradient exceeds a threshold. As the size of the eye region increases the number of operations increases by a factor of almost  $n^2$  where  $n$  is the number of pixels of the eye area while in our method this factor is  $n$  as it is shown in Figure 15(right).

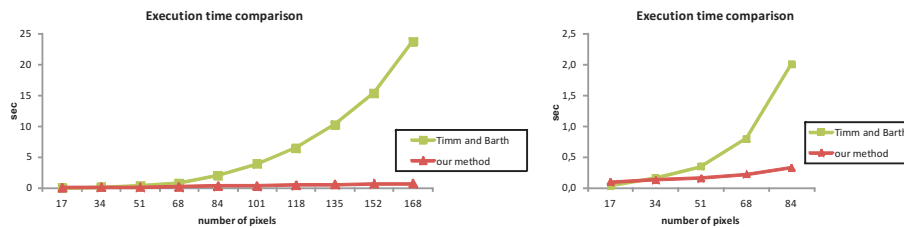


Fig. 15. Execution times comparison of our method with that of Timm and Barth as function of eye region size.

## 4.2 Corner detection

The proposed corner detection algorithm is applied to the two databases, i.e. BioID and GI4E. Compared to iris center detection, the evaluation of corner detection methods is more difficult since not many results have been published. Reasons for that could be, firstly, that the corner detection is not as relevant as iris center detection and, secondly, that the marks for the outer eye corner can vary between different databases due to the fact that there is not a unique definition for the outer eye corner. From the marks provided for the BioID database we deduce that the end of the sclera (white part of the eyeball) is labeled as the corner while in GI4E database it is the intersection between the upper and lower eyelids what is labeled as the eye corner. The distance between these two eye features can be negligible for some subject while it can be significant for many others.

Regardless of this issue, our algorithm was applied to both databases and the performance was measured. The average error between the eyes is normalized with the distance between the centers of the irises.

We measure the performance for the BioID database at the error value of 0.05. In order to simplify the analysis,  $e_{avg}$  is considered as measure of performance. Once the algorithm is applied to the database images, a clear offset in the horizontal coordinate of the corner is detected. As mentioned before, this is due to the fact that the BioID labeling selects the end of the sclera as the corner of the eye, while the proposed algorithm estimates the corner as the end point of the eyelid curve, which is slightly displaced from the sclera. In order to make a fair analysis of the data, this offset is partially compensated. An accurate correction of the offset is not possible since it depends on the user, thus an average correction is performed for the horizontal coordinate. Once this offset is corrected, the performance is approximately 65% (34% before correction), as shown in Figure 16. If the most difficult images of this

type, i.e. users with strong reflections in the glasses, are left out, the performance increases to 68%. The amount of images removed from the analysis represents 7% of the total.

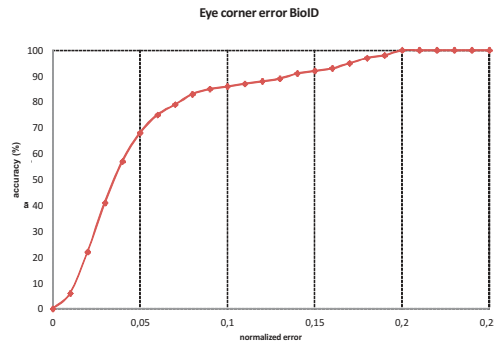


Fig. 16. Performance curve for outer eye corner detection algorithm as the mean result for the BioID database.

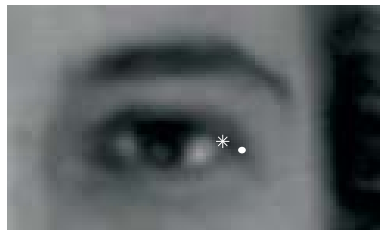


Fig. 17. Corners according to the labeling procedure for the BioID database and as result of our algorithm. The dot represents the point calculated by the corner detection algorithm, while the asterisk is the point according to the labels provided. Depending on the applications both points could be considered acceptable. If both of them are considered as correct the error threshold could be increased to 0.065%.

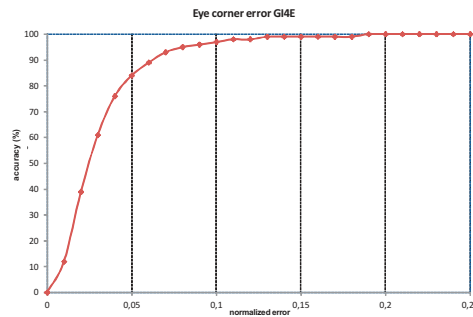


Fig. 18. Performance curve for corner detection algorithm as the mean result for the GI4E database.

It can be discussed if the value of 0.05 is a valid threshold to measure the performance of the corner detection, as it was for the iris. Since the definition of the corner is more vague compared to the iris,





Fig. 19. Cropped images of correct detections for samples of the BioID and GI4E databases.



Fig. 20. Cropped images of non correct results for samples of the BioID and GI4E databases.

different thresholds can be valid depending on the application. For the image of the BioID shown in Figure 17, the corners according to the labeling process and as result of our algorithm are sketched. For the specific image shown, both points could be acceptable. The error for this image is 0.065. The performance of the system at 0.065 is about 76%.

For the GI4E database the performance is shown in Figure 18. If 0.05 is selected as performance threshold, the performance reaches 84%. The increment in the performance with respect to the BioID was expected due to the higher quality of the images. The problems encountered are similar to the ones found for the BioID, such as users with strong reflections in the glasses. If the most difficult subjects are eliminated (about 6.5% of the images) the performance rises to 91%.

Figures 19 and 20 show correct and non correct estimations respectively for images of the BioID and GI4E databases. In Figure 20, errors for users wearing glasses with strong reflections in the eyelid area can be observed. However, the algorithm can overcome this problem in similar complex images as shown in Figure 19.

The work by Haiying and Guoping [2009] is selected for comparison since it is the one that resembles more closely our evaluation method. Their method to obtain the outer and inner corners results in an

average performance of 94% for the BioID database; however, they only employ 500 images from the database that are not clearly identified. The comparison is not completely valid since our method is applied over a higher number of images (close to 1500). As comparison, if the 500 images from the BioID with the best result are selected according to our method, the performance obtained is 100% at 0.05 (in fact the 100% is achieved at 0.03).

## 5. CONCLUSIONS

Iris center and eye corner detection is essential in several applications such as gesture recognition and low cost eye tracking among others. In this paper a novel iris center detection algorithm is proposed. We also suggest a robust and accurate method to detect outer corner of the eye. The method to detect the iris center is based on pit detection, according to image topography, at different resolution levels and works in real time. The most stable valley in the three resolutions is considered to be the iris center. A similar approach is employed to detect the outer corner of the eye. Since the eye corner is a less stable valley, the eyelid curve is used to strengthen its estimation. Two databases have been used to test our algorithms, i.e. BioID and GI4E. Regarding the iris center, for the BioID, our algorithm's performance is comparable to that of other existing algorithms, and it has the best average performance. BioID is the reference framework with which most of the methods in this field are compared. The sensitivity of our algorithm to uncorrect eye regions disappears and its performance improves significantly when higher-quality images are used, such as the ones provided by standard web cams, contained in the GI4E database. In this case, our method presents slightly better results than Timm and Barth [2011], which is one of the best algorithms for iris detection. On the other hand, our method clearly outperforms the work based on topography proposed by Wang et al. [2007] for iris center detection when more challenging images are tested. Regarding the evaluation of eye corner detection, both databases have been used. The performance for the BioID database is 76%, while it rises to 84% for GI4E dataset. The algorithm proposed in this paper presents the best mean performance, being the most robust in the field. Furthermore, it is faster than the closest algorithm. Our algorithm is simple since it does not use any training or learning stages. It is completely general since no parameters have to be adjusted between users and databases. Compared to other works which limit the experiments to BioID database, we have demonstrated the performance of the method in two different databases with significant differences regarding image quality. GI4E is publicly available by contacting the authors.

## Acknowledgment

Spanish Ministry of Science and Technology has supported this work under Contract TIN2009-12247.

## REFERENCES

- ASADIFARD, M. AND SHANBEZADEH, J. 2010. Automatic adaptive center of pupil detection using face detection and cdf analysis. In *IMECS*. Vol. I. Newswood Limited, 130–133.
- ASTERIADIS, S., NIKOLAIDIS, N., HAJDU, A., AND PITAS, I. 2006. An eye detection algorithm using pixel to edge information. In *2nd ISCCSP*. EURASIP.
- BAILENSON, J. N., PONTIKAKIS, E. D., MAUSS, I. B., GROSS, J. J., JABON, M. E., HUTCHERSON, C. A. C., NASS, C., AND JOHN, O. 2008. Real-time classification of evoked emotions using facial feature tracking and physiological responses. *Int. J. Hum.-Comput. Stud.* 66, 303–317.
- BEHNKE, S. 2002. Learning face localization using hierarchical recurrent networks. In *ICANN*. 1319–1324.
- BELHUMEUR, P. N., JACOBS, D. W., KRIEGMAN, D. J., AND KUMAR, N. 2011. Localizing parts of faces using a consensus of exemplars. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- BERTOLINO, P. AND MONTANVERT, A. 1996. Multiresolution segmentation using the irregular pyramid. In *In R.J. Greenspan and C.P. Kyriacou, editors, Flexibility and Constraint in Behavioral Systems*. John Wiley and Sons, 257–260.
- BICEGO, M., LAGORIO, A., GROSSO, E., AND TISTARELLI, M. 2006. On the use of sift features for face authentication. *CVPRW '06*. IEEE Computer Society, Washington, DC, USA, 35–.
- BOLT, R. A. 1982. Eyes at the interface. In *Proceedings of the 1982 conference on Human factors in computing systems*. CHI '82. ACM, New York, NY, USA, 360–362.
- CAMPADELLI, P. AND LANZAROTTI, R. 2006. Precise eye localization through a general-to-specific model definition.
- CHEN, D., TANG, X., OU, Z., AND XI, N. 2006. A hierarchical floatboost and mlp classifier for mobile phone embedded eye location system. In *ISNN (2)*. 20–25.
- CRISTINACCE, D., COOTES, T., AND SCOTT, I. 2004. A multi-stage approach to facial feature detection. In *15<sup>th</sup> British Machine Vision Conference, London, England*. 277–286.
- DIBEKLIOGLU, H., SALAH, A. A., AND GEVERS, T. 2011. A statistical method for 2d facial landmarking. *IEEE Transactions on Image Processing*.
- DYER, C. R. 1987. *Parallel computer vision*. Academic Press Professional, Inc., San Diego, CA, USA, Chapter Multiscale image understanding, 171–213.
- ELLIS, S., CANDREA, R., MISNER, J., CRAIG, C. S., AND LANKFORD, C. P. 1998. Windows to the Soul? What Eye Movements Tell Us About Software Usability. *Proceedings of the 7th Annual Conference of the Usability Professionals' Association*.
- FERDOWSI, S. AND AHMADYFARD, A. 2008. Using statistical moments as invariants for eye detection. In *16th European Signal Processing Conference (EUSIPCO 2008)*. EURASIP, Laussane, Switzerland.
- FUKUDA, T., MORIMOTO, K., AND YAMANA, H. 2010. Model-based eye-tracking method for low-resolution eye-images. In *International Workshop on Eye Gaze in Intelligent Human Machine Interaction, February 2010, Hong Kong, China*.
- HAIYING, X. AND GUOPING, Y. 2009. A novel method for eye corner detection based on weighted variance projection function. In *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*. 1–4.
- HAMOUIZ, M., KITTLER, J., KAMARAINEN, J. K., PAALANEN, P., KALVIAINEN, H., AND MATAS, J. 2005. Feature-based affine-invariant localization of faces. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1490–1495.
- HANSEN, D. W. AND JI, Q. 2010. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 3, 478–500.
- INCE, I. AND KIM, J. 2011. A 2d eye gaze estimation system with low-resolution webcam images. *EURASIP J. Adv. Sig.Proc.*
- INCE, I. F. AND YANG, T.-C. 2009. A new low-cost eye tracking and blink detection approach: extracting eye features with blob extraction. *ICIC'09*. Springer-Verlag, Berlin, Heidelberg, 526–533.
- JESORSKY, O., KIRCHBERG, K. J., AND FRISCHHOLZ, R. 2001. Robust face detection using the hausdorff distance. In *AVBPA*. 90–95.
- KROON, B., HANJALIC, A., AND MAAS, S. M. P. 2008. Eye localization for face matching: is it always useful and under what conditions? In *CIVR*. 379–388.
- LINDBERG, T. 1994. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 224–270.
- MAJARANTA, P., AOKI, H., DONEGAN, M., HANSEN, D. W., HANSEN, J. P., HYRSKYKARI, A., AND RAIHA, K. 2011. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies*. IGI Global.
- MEER, P. AND WEISS, I. 1992. Smoothed differentiation filters for images. *J. Vis. Comun. Image Represent.* 3, 58–72.
- MERCHANT, J., MORRISSETTE, R., AND PORTERFIELD, J. 1974. Remote measurement of eye direction allowing subject motion over one cubic foot of space. *IEEE Transactions on Biomedical Engineering* 21, 4, 309–317.

- MIN, R., HADID, A., AND DUGELAY, J.-L. 2011. Improving the recognition of faces occluded by facial accessories. In *FG 2011, 9th IEEE Conference on Automatic Face and Gesture Recognition, March 21-25, 2011, Santa Barbara, CA, USA*. Santa Barbara, UNITED STATES.
- MITRA, S. AND ACHARYA, T. 2007. Gesture recognition: A survey. *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS - PART C* 37, 3, 311–324.
- MONTY, R., SENDERS, J., AND LABORATORY, U. A. H. E. 1976. *Eye movements and psychological processes*. Lawrence Erlbaum Associates.
- MORIMOTO, C. H. AND MIMICA, M. R. M. 2005. Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.* 98, 4–24.
- MORIYAMA, T., KANADE, T., XIAO, J., AND COHN, J. F. 2006. Meticulously detailed eye region model and its application to analysis of facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 738–752.
- NIU, Z., SHAN, S., YAN, S., CHEN, X., AND GAO, W. 2006. 2d cascaded adaboost for eye localization. In *ICPR (2)*. 1216–1219.
- PHILLIPS, P. 1998. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing* 16, 5, 295–306.
- PONG, T.-C., SHAPIRO, L. G., AND HARALICK, R. M. 1985. Shape estimation from topographic primal sketch. *Pattern Recognition* 18, 5, 333–347.
- PONZ, V., VILLANUEVA, A., SESMA, L., ARIZ, M., AND CABEZA, R. 2011. Topography-based detection of the iris centre using multiple-resolution images. In *Irish Machine Vision and Image Processing Conference (IMVIP 2011) September 2011*. 1–4.
- POOLE, A. AND BALL, L. J. 2005. Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. In *Encyclopedia of Human Computer Interaction*, C. Ghaoui, Ed. IGI Global.
- RESEARCH, B. T. 2001. The bioid face database. [www.bioid.com](http://www.bioid.com).
- SEWELL, W. AND KOMOGORTSEV, O. 2010. Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. CHI EA '10. ACM, New York, NY, USA, 3739–3744.
- SHIH, F. Y., CHUANG, C.-F., AND WANG, P. S. P. 2008. Performance comparisons of facial expression recognition in jaffe database. *IJPRAI* 22, 3, 445–459.
- SIGUT, J. AND SIDHA, S.-A. 2011. Iris center corneal reflection method for gaze tracking using visible light. *Biomedical Engineering, IEEE Transactions on* 58, 2, 411–419.
- STARKER, I. AND BOLT, R. A. 1990. A gaze-responsive self-disclosing display. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*. CHI '90. ACM, New York, NY, USA, 3–10.
- TAHERI, S., TURAGA, P., AND CHELLAPPA, R. 2011. Towards view-invariant expression analysis using analytic shape manifolds. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE.
- TIAN, Y.-L., KANADE, T., AND COHN, J. F. 2000. Eye-state action unit detection by gabor wavelets. In *Proceedings of the Third International Conference on Advances in Multimodal Interfaces*. ICMI '00. Springer-Verlag, London, UK, 143–150.
- TIMM, F. AND BARTH, E. 2011. Accurate eye centre localisation by means of gradients. In *Proceedings of the Int. Conference on Computer Theory and Applications (VISAPP)*. Vol. 1. INSTICC, Algarve, Portugal, 125–130.
- TÜRKAN, M., PARDÀS, M., AND ÇETIN, A. E. 2007. Human eye localization using edge projections. In *VISAPP (1)*. 410–415.
- VALENTI, R. AND GEVERS, T. 2008. Accurate eye center location and tracking using isophote curvature. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- VALENTI, R., STAIANO, J., SEBE, N., AND GEVERS, T. 2009. Webcam-based visual gaze estimation. In *ICIAP*. 662–671.
- VERTEGAAL, R. 1999. The gaze groupware system: mediating joint attention in multiparty communication and collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. CHI '99. ACM, New York, NY, USA, 294–301.
- VIOLA, P. AND JONES, M. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57, 137–154.
- WANG, J., YIN, L., AND MOORE, J. 2007. Using geometric properties of topographic manifold to detect and track eyes for human-computer interaction. *TOMCCAP* 3, 4.
- XU, C., ZHENG, Y., AND WANG, Z. 2008. Semantic feature extraction for accurate eye corner detection. In *19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA*. IEEE, 1–4.
- ZHOU, R., HE, Q., WU, J., HU, C., AND MENG, Q. H. 2011. Inner and outer eye corners detection for facial features extraction based on ctgf algorithm. *Applied Mechanics and Materials Volume Information Technology for Manufacturing Systems II*.
- ZHOU, Z.-H. AND GENG, X. 2004. Projection functions for eye detection. *Pattern Recognition* 37, 5, 1049–1056.
- ZHU, J. AND YANG, J. 2002. Subpixel eye gaze tracking. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. 124–129.