Journal of Eye Movement Research
7(3):3, 1-8

Evaluation of accurate eye corner detection methods for gaze estimation

Jose Javier Bengoechea
Public University of Navarra, Spain

Juan J. Cerrolaza
Childrens National Medical Center, USA

Arantxa Villanueva
Public University of Navarra, Spain

Rafael Cabeza
Public University of Navarra, Spain

Accurate detection of iris center and eye corners appears to be a promising approach for low cost gaze estimation. In this paper we propose novel eye inner corner detection methods. Appearance and feature based segmentation approaches are suggested. All these methods are exhaustively tested on a realistic dataset containing images of subjects gazing at different points on a screen. We have demonstrated that a method based on a neural network presents the best performance even in light changing scenarios. In addition to this method, algorithms based on AAM and Harris corner detector present better accuracies than recent high performance face points tracking methods such as Intraface.

Keywords: eye tracking, low cost, eye inner corner

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 as shown in Morimoto and Mimica (2005) and gesture recognition as described by Tian, Kanade, and Cohn (2000) and Bailenson et al. (2008).

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 as shown in the work by Merchant, Morrissette, and Porterfield (1974). Since then, and especially in the last decades, much effort has been devoted to improving the performance of eye tracking systems. The availability of high performance eye tracking systems has provided advances in fields such as usability research as described by Ellis, Candrea, Misner, Craig, and Lankford (1998) Poole and Ball (2005) and interaction for severely disabled people in works such as Bolt (1982), Starker and Bolt (1990) and 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 to interact 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. 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 specialized 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 (Zhang, Bulling, & Gellersen, 2013). However, simpler (i.e., lower cost) hardware is needed to reach these areas.

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 as described in 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. 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 eye-

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692

ball size is directly proportional to the deviation of cursor movements in a certain screen size and resolution. Fukuda, Morimoto, and Yamana (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, Staiano, Sebe, and Gevers (2009) and in Sesma, Villanueva, and Cabeza (2012). The work by Zhu and Yang (2002) presents a web cam 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.

Compared to other facial features detection methods accuracy is key for gaze estimation purposes. Recently, several papers have been presented about accurate iris center detection using a web cam as shown in Timm and Barth (2011) and Villanueva et al. (2013), however, not much has been published about accurate eye corner detection. Regarding eye corner estimation, we find works in which corners are detected as a result of facial features detection methods. Recently, Dibeklioglu, Salah, and Gevers (2011) and Belhumeur, Jacobs, Kriegman, and Kumar (2011) have presented relevant works in the area. 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, He, Wu, Hu, and Meng (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. However, none of this methods present the require accuracy for gaze estimation purposes.

In this paper, we propose a group of novel eye inner corner detection methods providing higher accuracies. In our previous work Villanueva et al. (2013), we suggested a method to detect the outer corner of the eye. However, recent experiments show that the inner corner shape is more robust and stable. Thus, we propose new methods for this corner detection. On the one hand, we adapt well-known appearance based segmentation methods for corner detection and on the other hand we use improved techniques for feature detection and eye corner segmentation.

This paper is organized as follows. In the next section appearance based segmentation methods are presented. To follow, features detection methods are described together with the initialization strategy employed. Finally, the experiments carried out and the results are presented.

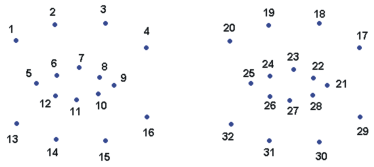

Figure 1. The points shown in the figure are the alternative landmarks that have to be placed by the expert in the eye area.

Figure 2. One of the training images in which the ASM model is overlapped.

Appearance based methods

Active Shape Models (ASM) and Active Appearance Models (AAM) methods have been largely used for segmentation of facial features. These detection techniques were introduced by Cootes, Edwards, and Taylor (2001) and Cootes, Taylor, Cooper, and Graham (1995) and are based on a previous learning procedure in which an expert marks key segmentation points in images from a training set. Both techniques study the statistical behavior of the object to be segmented in terms of shape, appearance and texture. ASM segmentation method learns the possible shapes of an object and the appearance of the landmarks while AAM is able to learn textures of the object in addition to the shape.

ASM and AAM learn the patterns of variability of a shape, estimating the population statistics from a set of training examples, i.e. a set of already segmented images. The shape of interest is defined in each image of the training set by the join-the-dots approach; that is, a set of landmarks distributed through the shape. It is important to notice that each landmark defines a point of correspondence between all the shapes, so they must be carefully placed in the image. From all this information, the ASM must learn the range of variation of the shape and how to automatically locate the landmarks that define it. While only shape constraints are defined in ASM, AAM also models the texture of the target object via PCA. Finally, the coupled relationship between the shape and the texture is analyzed to create a combined final model.

A general description of the shape model is presented below. This model is presented in both ASM and AAM algorithms, allowing us to illustrate the main ideas of PCA based statistical models. Because a detailed description of these methods is out of the scope of this text, the reader is advised to consult the cited references for further information. The statistical shape model is built by computing a PDM (Point Distribution Model) from the set of shapes.

Applying Principal Component Analysis (PCA) to

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692

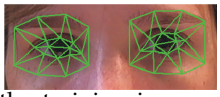

Figure 3. One of the training images in which the AAM model is overlapped.

the data, we can obtain the modes that best describe the observed variation, which allows us to reduce the dimensionality of the data. In this way, any instance of the shape space can be approximated by the equation

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \tag{1}$$

where $\mathbf{b} = (b_1, b_2, ..., b_t)^T$ is a t dimensional vector defining the set of parameters of the statistical shape and appearance model; $\mathbf{P} = (\mathbf{p}_1 \mid \mathbf{p}_2 \mid ... \mid \mathbf{p}_t)$ is a matrix containing the t main eigenvectors of the covariance matrix, that is, those associated to the t highest eigenvalues.

Typically, the number of modes to retain, t, is chosen as a proportion f_v (e.g. $95-98\%$) of the total variance exhibited in the training set we wish to explain; although the use of alternative approaches is also possible (Cootes & Taylor, 2004). The total variance is defined by the sum of all the eigenvalues, λ_i; thus t can be easily obtained as the minimum integer that satisfies $\sum_{i=1}^{t} \lambda_i \geq f_v \sum_{\forall i} \lambda_i$.

Although more sophisticated alternatives have been proposed (Stegmann, Fisker, & Ersbøll, 2000) (Cerrolaza, Villanueva, & Cabeza, 2011), one of the simplest and most popular ways to guarantee that only legitimate instances are generated by equation (1) is to constraint $\mathbf{b}$ to lie inside an hyperrectangle by applying hard limits to each element, $\mid b_i \mid \leq \beta\sqrt{\lambda_i}$ where β is a constant (typically between 1 and 3) that determines the flexibility of the model.

The appearance model of each landmark must also be extracted from the training set, building the statistical model of fixed-size grey value profiles, normal to the boundary of the object and centered at each landmark. In order to reduce the effect of global intensity changes, the normalized first order derivative of the grey level profile is used. This image-driven update of landmarks is alternated with a shape adjustment step, creating an iterative segmentation process.

In this paper, we have tested a variety of models based on ASM and AAM in order to detect the eye inner corner. The objective is to segment a facial characteristic, e.g. eye contour, that includes the eye inner corner as landmark. Thus, once the whole facial feature is segmented the landmark of the model corresponding to the eye inner corner is provided as result.

The two proposed methods try to model the area containing both eyes. The employed landmarks are shown in figure 1. The first method is based on ASM and tries to model the behavior of the shape shown in figure 2. The model contains both eyes and landmarks number 9 and 25 are the ones corresponding to the inner corner of both eyes. We refer the reader to the paper Cootes et al. (1995) including more details about the implementation used for the segmentation.

The second method modelling the eye area is based on AAM. The facial area modelled is shown in figure 3. Once the landmarks have been established AAM is able to model the texture of the object areas contained in the triangles shown in the figure that are constructed using the landmarks as reference.

Both ASM and AAM models have been trained using 83 images of different subjects gazing at alternative points on the screen. One of the weaknesses of ASM and AAM is their sensitivity to the initialization. The model obtained from the training stage needs to be placed close to the real contour of the new instance of the object. If the initialization is appropriate, the model is adapted to the shape of the object in the image by means of an iterative procedure that modifies the location of the landmarks according to the learnt model. Both methods, i.e. ASM and AAM, are initialized using the position of the iris centers of the eyes. Once the position of the iris is known as described in Villanueva et al. (2013), its position in the image is used to initialize the model.

Feature detection methods

Three methods are proposed in order to detect eye inner corner. The first method is based on Harris corner detection, the second method uses Canny edge detection in order to detect the eyelid. Both methods are applied in a previously estimated searching area in which the eye corner is contained. The detection of this region is based on a neural network stage. The third method detects the eye corner as a result of a postprocessing stage of the area provided by the neural network. The detection of this region of interest using neural networks is explained first.

In order to establish this searching area alternative methods have been tested. Using the irises of the eye is one of the evaluated options, i.e. selecting a window between the iris center and the nose. However, it presents a low robustness since the iris center position can vary as a function of gaze direction.

The method proposed is a multi-stage procedure in which the face is previously segmented using Viola-Jones face detector. The segmentation provides a rectangle in which the face is contained. Thus, the eye area is segmented assuming that the eyes will be contained in the upper part of the face. Left and right eyes areas are obtained by dividing the eye area in two parts.

A neural network has been trained to detect the eye inner corner area (in the previously segmented eye areas). The network employed is a feed-forward back propagation network with one hidden layer. We use RGB patches with size 7x7 of the eye area to train the neural network. In order to achieve the best results, we train the network using more negative than positive

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692

examples of the eye area, thus, we employ 20% more negative than positive examples.

Once the neural network has been trained it is applied in the eye area. For each one of the methods proposed requiring to be initialized the neural network provides a reduced searching area with more that 90% of certainty.

The Harris corner detector is a popular interest point detector due to its strong invariance to rotation, scale, illumination variation and image noise. The Harris corner detector measures the local changes of the signal with patches shifted by a small amount in different directions. The aim is to find little patches of the image that generate a large variation when moved around. Given a pixel and its neighborhood a matrix M can be calculated as:

$$M = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \quad (2)$$

where, I_x and I_y are the derivatives of the image in x and y directions. A score R can be calculated for each point given as $R = det(M) - k * trace(M)$ where $det(M) = \lambda_1 \lambda_2$ and $trace(M) = \lambda_1 + \lambda_2$ and λ_1 and λ_2 are the eigenvalues of matrix M.

The method proposed applies a modified version of the Harris corner detector in the previously detected searching area. The Harris corner detector is applied in the original window and in a smoothed version of the same window with $k = 0.06$. The searching subimage is smoothed using a Gaussian filter. According to the size of our test images, the filter size is 19x19 and $\sigma = 4$. In the original window all the local maxima of the score R are detected as candidates, while a threshold is established in the smoothed image in order to classify it as a candidate. This threshold has been fixed as 50% of the maximum value of R. As expected, less corners are detected in the smoothed version of the image (see figure 4). From our experience in our test images the number of images in which more than one corner are detected (the maximum is two) is negligible in comparison to the total number of images tested. Nevertheless, in those cases the candidate closer to the nose is selected as the best approximation of the corner in the smoothed image.

In the original searching window the Harris detector segments all local maxima as corner candidates. The number of candidates makes it difficult to select the one corresponding to the inner corner. However, the detections present higher accuracy especially if this is compared with the one obtained in the smoothed version of the image.

The method proposed combines the detections obtained in the original and smoothed versions of the searching window. Thus, the inner corner is selected as the candidate in the original window closer to the one obtained in the smoothed image. In this manner, the accuracy obtained in the original window is combined with the predictive ability of the Harris method in the low resolution version of the image.

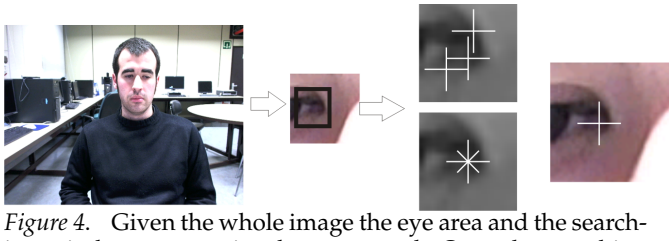

Figure 4. Given the whole image the eye area and the searching window are previously segmented. Once the searching area is applied the eye inner corner is calculated as the corner obtained in high resolution image (cross) closer to the corner obtained in the lower resolution image (asterisk).

The second method proposed combines two well known techniques to detect the inner corner of the eye. It combines an edge detection method and topography to determine the eye inner corner. The searching area is also previously calculated as it has been previously described in this section. In the searching area we use a Canny edge detector to segment the upper eyelid. For the test images σ=2 is used. In addition, since the upper eyelid has a stronger horizontal component only the corresponding component of the Canny detector is used, i.e. the horizontal component. Once the edge detection is performed, we apply an algorithm to detect curves, which basically fills gaps between edge segments to reconstruct the curve. We consider that two edge segments belong to the same curve when their distance is one pixel. Thus, the point of this curve closer to the nose is considered to be a good approximation of the eye inner corner (see figure 5). Our experience shows that the detection of this point is robust but it only approximates the real position of the eye inner corner. The main reason for that is the fact that in most of the cases the eyelid is not completely segmented. Nevertheless, it is assumed that this point is a good approximation of the eye inner corner.

In order to improve the accuracy of the method we introduce pit detection. From image topography theory perspective, image pixels can be labeled according to their grey level and the intensity of their neighbouring pixels (Wang, Yin, & Moore, 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:

$$\mathbf{H}(x,y) = \begin{pmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{pmatrix} \quad (3)$$

From the eigenvalue decomposition of $\mathbf{H}$, λ_1 and λ_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 & Weiss, 1992). Image topography allows the labeling of pixels as ridge

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692

Figure 5. The method based on Canny edge detector is shown. The upper eyelid is previously segmented and the pits in the area are calculated (white dots). The pit selected as the eye inner corner is the one closer to the end point of the eyelid.

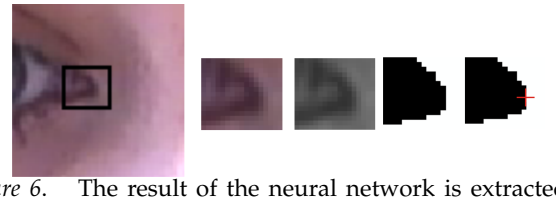

Figure 6. The result of the neural network is extracted as a subimage. This subimage is converted to gray levels, it is thresholded using Otsu's method. The point closer to the nose is selected as the pixel representing the eye inner corner.

and pits among others. Thus, the eye corner 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 $\|\nabla f(x,y)\| = 0, \lambda_1 > 0, \lambda_2 > 0$.

According to our results, the eye corner is classified as a pit under image topography perspective. The advantage of topography compared to eyelid segmentation is its accuracy. Thus, we select as inner corner the pit closer to the end point of the eyelid segmented previously.

The last method proposed performs a post processing of the image area provided by the neural network. The image area provided by the network is extracted as a new image. This image patch is first converted to gray levels. Secondly, it is thresholded using Otsu's method (see figure 6). Otsu's method is an automatic thresholding method based on Bayesian classification methods (Otsu, 1979). The algorithm considers that the image to be thresholded contains two objects represented by two classes of pixels, i.e. foreground pixels and background pixels. The algorithm consists in calculating the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal which is equivalent to maximizing the inter-class variance. The inter-class variance is defined as: $\sigma_B^2(k) = P_1(k)P_2(k)(m_1(k) - m_2(k))^2$ where $P_i(k)$ is the probability of a pixel to belong to class i and $m_i(k)$ is the mean intensity value of the pixels assigned to class i. All these magnitudes depend on the selected threshold k. Otsu's methods is based on finding the value of k that maximizes σ_B^2. Once the threshold value is automatically selected a binary image is obtained. The corner of the eye is calculated as the point closer to the nose labeled as zero.

Experiments

We have tested five algorithms over images taken from the proprietary GI4E dataset. 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. It contains high resolution images of more than 100 subjects gazing at different points in the screen, resulting in more than 1200 images. The dataset consists of colour images from different subjects aged from 18 to 83 years old, males and females. The subjects were asked to sit in front of a computer and to look at 12 different points uniformly distributed on the screen, to simulate the conditions that take place when using a computer in an ordinary way. Thus, the database contains images of people looking at different directions, which makes this set different from the most commonly used datasets for eye detection, in which the subjects look mainly at the camera. The images were taken indoors at different locations and at different day times, which results in variable backgrounds and illumination conditions. There are position changes as well, since the subjects behaved in their natural way while the images were taken. Some of the subjects wear glasses and their eyes are hidden by reflections, while others have hair near the eyes, causing occlusions.

The set of images was acquired using a low cost web cam, with automatic lighting correction, and the image size is 800x600 pixels. No other equipment or specific illumination such as infrared was used in image acquisition. Each image has been annotated by three different people, labelling the centre of the iris, and the inner and outer corners of the eyes. The final annotations are calculated as the average of the three marks, stored in a text file and provided with the images.

We have selected 200 images from the database arbitrarily. In these images the inner corner has been remarked by two experts more accurately. The error of the algorithms is calculated as the distance between the estimated inner corner position and the real position as provided in the label. This error is normalized with respect to the distance between the irises of the eye.

$$e = \frac{\| corner_{est} - corner_{label} \|}{iris_{distance}} \tag{4}$$

Results

The graph in figure 7 shows the cumulative error for each one of the algorithms, i.e. the horizontal axis indicates error values, i.e. e, and the vertical one the percentage of images for which the error in the corner detection is below that error threshold e. Apart from the methods presented in the paper, we have considered appropriate to include a novel high impact method

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692

in the comparison, i.e. Intraface. It tracks selected face features in which eye corners are included. Basically, it proposes a supervised Descent Method (SDM) for minimizing a Non-linear Least Squares (NLS) function. During training, the SDM learns a sequence of descent directions that minimizes the mean of NLS functions sampled at different points (Xuehan-Xiong & De la Torre, 2013).

From figure 7 it can be observed that the method based on post processing the neural network result has the best performance. We have measured the variability of the marks created by the experts marking the database. The standard deviation of the inner corner marks in the database represents a limit, hence this error value would be the best performance an algorithm could obtain. The standard deviation is normalized with respect to the distance between the iris. For our database the value obtained is ~ 0.004 very similar to the one obtained for the iris center. From our experience error value of 0.05 can be acceptable for iris center detection. In this case, all the methods proposed are above 80% of performance, i.e. the 80% of the images present errors below 0.05 in the detected corners. However, our results have shown that the accuracy in the eye inner corner is more critical for gaze estimation. If accuracies of 0.01-0.02 are required the difference between the method based on thresholding the subimage resulting from the neural network presents an outstanding difference with respect to the rest of the methods. Furthermore, the difference of AAM and Harris with respect to the other two methods, i.e. Canny and ASM, is also significant. While AAM and Harris remain above 80% the performance of Canny and ASM decrease below 50% for error values of 0.02. As mentioned before the best method based on post processing the output of the neural network presents a performance of about 90% in the range of 0.01 and 0.02. As expected the Intraface method presents and acceptable behavior but its performance remains below three of our methods, i.e. AAM, Harris and the method based on thresholding the output of the neural network. A potential drawback of the method thresholding the output of the neural network is that it is highly dependent on the output of the network, i.e. a little change in the window can modify the threshold calculated by Otsu's method distorting the shape of the objects after thresholding. In contrast, the methods based on Canny and Harris can somehow compensate slight errors of the neural network window by imposing additional constraints. In our experiments no errors for the neural network have arisen, however, it is widely known that the performance of a neural network is highly dependent on the training set. In our case, a careful selection of the training examples was carried out. If new kind of images, i.e. different lighting setups, image quality... are tested, critical results can be obtained. For new type of images, a more robust behavior of the methods using Canny and Harris can be predicted. In this case, further studies and training stages would be required to adjust the neural network.

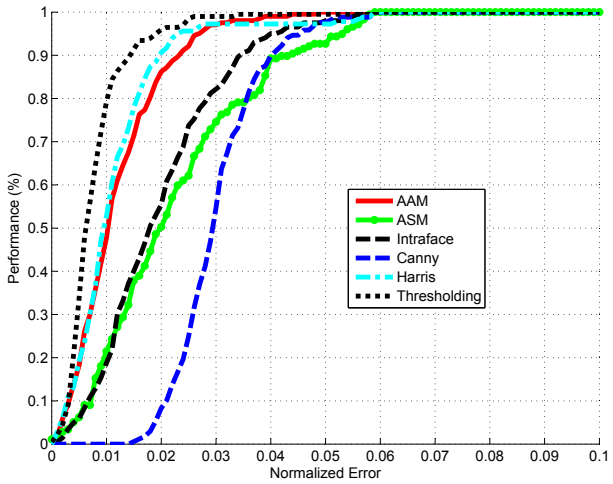

Figure 7. The horizontal axis shows the normalized error of eye inner corner estimation while the vertical shows the performance for each one of the methods. The performance is measured as the percentage of images below certain error value.

Computational cost

Improving the execution times of the techniques is not the objective of this work. The platform employed is a Windows XP x64 system performing at 3.3 GHz and 8 GB of RAM. The hardware employed and the software coded in Matlab have not been adapted to achieve the best performance in terms of speed. However, we find it appropriate to provide some numbers about the performance of each one of the proposed techniques. As expected, appearance based models present the highest numbers, moreover the execution times of feature detection methods is negligible compared to ASM and AAM. On the other hand, Canny and Harris detectors present a comparable behavior. The execution times of ASM and AAM are 171% and 285%, higher than the ones achieved by feature detection methods. In addition, regarding the Canny and Harris methods the 99% of the execution time is devoted to calculating the searching area employing the neural network. Once the searching window is determined, the time employed by feature detection methods is of milliseconds. In that sense, the method based on thresholding the neural network resulting area is the fastest one, since the additional thresholding computation time is negligible compared to the ones required for Canny and Harris methods. From the results obtained real time performance for the method based on

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692

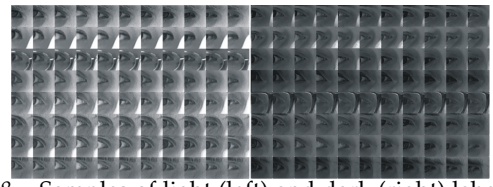

Figure 8. Samples of light (left) and dark (right) labeled images.

thresholding the neural network result can be assured in a state of the art platform once the software platform is improved and more real time oriented programming language is used such as C. Performance data for Intraface method has been obviated in this section since no data about the programming language of the functions used internally are provided. It has been also assumed that the Intraface method has been improved in terms of execution times and this was not the main objective of our work for the proposed methods.

Robustness to lighting conditions

In this section the performance of the different methods when the lighting conditions are varied has been measured. As it has been mentioned before, GI4E database contains images for which the light changes. In order to test this fact in a systematic manner, two groups of 100 images each have been selected classified as low and high light conditions (see figure 8). All the methods have been tested using these two sets of images.

All the methods present changes in their results when light conditions are modified, however, none of them suffers a significant improvement or diminishment in its performance, i.e. all of them present values above 80% for the error value of 0.05. Not all the methods are equally affected when the light varies, however, we can conclude that all of them present acceptable behavior in changing light scenarios. The main conclusion is that the method based on thresholding the output of the neural network remains as the best method in both conditions. In figure 9 we can observe that its performance is slightly worse for the group of dark images.

Conclusions

Low cost gaze estimation using web cams presents new challenges for researchers working in the field. If no infrared light sources are used iris center and eye corners can be considered as valid features for gaze estimation. Most of the algorithms devoted to detecting iris and eye corners lack of the required accuracy for gaze estimation purposes. We propose five novel approaches for eye inner corner accurate detection. Two of the methods proposed are methods based on appearance while the other three are methods for features detection.

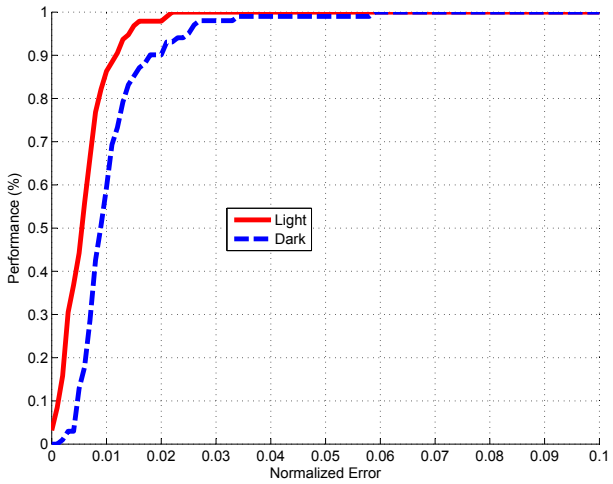

Figure 9. Performance of the best method based on thresholding the output of the neural network when light conditions are varied. The behavior is slightly better in high light conditions but both curves represent acceptable behaviors.

The methods have been tested over 200 images in which the eye inner corner has been accurately marked. The results show that for our experimental setup the method based on postprocessing the neural network output area presents the best performance. There is not a clear difference between the methods based on appearance and the methods for features detection. In fact, the method based on AAM and the method based on Harris detector present similar behaviors. If no additional requirements of the system are provided both types methods may be valid. The methods presented in this paper require training stages. If the training set is not carefully selected the methods based on ASM, AAM, and the one thresholding the neural network output can retrieve wrong results. Furthermore, critical situations can arise if new type of images are tested. We can expect a more robust behavior of the methods based on Canny and Harris algorithms since additional constraints are imposed. The training stage may represent a problem in certain applications and can represent a drawback of these methods. If this is not the case, our results show that the training of a neural network is more effective compared to ASM and AAM. In addition, the training is simpler since no landmarks are required and the execution times shorter. Finally, the Intraface method has been tested and proved to be in the range of the methods presented in this paper.

References

Bailenson, J. N., Pontikakis, E. D., Mauss, I. B., Gross, J. J., Jabon, M. E., Hutcherson, C. A. C., ... John, O. (2008,

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692

May). Real-time classification of evoked emotions using facial feature tracking and physiological responses. *Int. J. Hum.-Comput. Stud.*, *66*, 303–317.

Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., & Kumar, N. (2011, June). Localizing parts of faces using a consensus of exemplars. In *The 24th ieee conference on computer vision and pattern recognition (cvpr)*.

Bolt, R. A. (1982). Eyes at the interface. In *Proceedings of the 1982 conference on human factors in computing systems* (pp. 360–362). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/800049.801811 doi: http://doi.acm.org/10.1145/800049.801811

Cerrolaza, J. J., Villanueva, A., & Cabeza, R. (2011). Shape constraint strategies: Novel approaches and comparative robustness. In *Proceedings of the british machine vision conference* (pp. 7.1–7.11).

Cootes, T., Edwards, G., & Taylor, C. (2001, JUN). Active appearance models [Article]. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, *23*(6), 681-685.

Cootes, T., & Taylor, C. (2004). *Statistical models of appearance for computer vision* (Tech. Rep.). Department of Imaging Science and Biomedical Engineering, University of Manchester.

Cootes, T., Taylor, C., Cooper, D., & Graham, J. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding*, *61*(1), 38 - 59.

Dibeklioglu, H., Salah, A. A., & Gevers, T. (2011). A statistical method for 2d facial landmarking. *IEEE Transactions on Image Processing*.

Ellis, S., Candrea, R., Misner, J., Craig, C. S., & 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*.

Fukuda, T., Morimoto, K., & 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., & Guoping, Y. (2009, oct.). 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* (p. 1 -4). doi: 10.1109/CISP.2009.5304434

Ince, I. F., & Yang, T.-C. (2009). A new low-cost eye tracking and blink detection approach: extracting eye features with blob extraction. In (pp. 526–533). Berlin, Heidelberg: Springer-Verlag.

Meer, P., & Weiss, I. (1992, March). Smoothed differentiation filters for images. *J. Vis. Comun. Image Represent.*, *3*, 58–72.

Merchant, J., Morrissette, R., & Porterfield, J. (1974, July). Remote measurement of eye direction allowing subject motion over one cubic foot of space. *IEEE Transactions on Biomedical Engineering*, *21*(4), 309-317.

Morimoto, C. H., & Mimica, M. R. M. (2005, April). Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.*, *98*, 4–24.

Otsu, N. (1979). A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, *9*(1), 62–66.

Poole, A., & Ball, L. J. (2005, December 30). Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. In C. Ghaoui (Ed.), *Encyclopedia of human computer interaction.* IGI Global.

Sesma, L., Villanueva, A., & Cabeza, R. (2012). Evaluation of pupil center-eye corner vector for gaze estimation using a web cam. In *Proceedings of the symposium on eye tracking research and applications* (pp. 217–220). ACM.

Sewell, W., & 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* (pp. 3739–3744). New York, NY, USA: ACM.

Sigut, J., & Sidha, S.-A. (2011, feb.). Iris center corneal reflection method for gaze tracking using visible light. *Biomedical Engineering, IEEE Transactions on*, *58*(2), 411 -419.

Starker, I., & Bolt, R. A. (1990). A gaze-responsive self-disclosing display. In *Proceedings of the sigchi conference on human factors in computing systems: Empowering people* (pp. 3–10). New York, NY, USA: ACM.

Stegmann, M. B., Fisker, R., & Ersbøll, B. K. (2000). *On properties of active shape models* (Tech. Rep.). Informatics and Mathematical Modelling, Technical University of Denmark, DTU.

Tian, Y.-l., Kanade, T., & Cohn, J. F. (2000). Eye-state action unit detection by gabor wavelets. In *Proceedings of the third international conference on advances in multimodal interfaces* (pp. 143–150). London, UK: Springer-Verlag.

Timm, F., & 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, pp. 125–130). Algarve, Portugal: INSTICC.

Valenti, R., Staiano, J., Sebe, N., & Gevers, T. (2009). Webcam-based visual gaze estimation. In *Iciap* (p. 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* (pp. 294–301). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/302979.303065 doi: http://doi.acm.org/10.1145/302979.303065

Villanueva, A., Ponz, V., Sesma-Sanchez, L., Ariz, M., Porta, S., & Cabeza, R. (2013). Hybrid method based on topography for robust detection of iris center and eye corners. *ACM Transactions on Multimedia Computing Communications and Applications*, in press.

Wang, J., Yin, L., & Moore, J. (2007). Using geometric properties of topographic manifold to detect and track eyes for human-computer interaction. *TOMCCAP*, *3*(4).

Xuehan-Xiong, & De la Torre, F. (2013). Supervised descent method and its application to face alignment. In *Ieee conference on computer vision and pattern recognition (cvpr)*.

Zhang, Y., Bulling, A., & Gellersen, H. (2013). Sideways: A gaze interface for spontaneous interaction with situated displays. In *Proceedings of the computer human interaction conference, chi '13.*

Zhou, R., He, Q., Wu, J., Hu, C., & 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*.

Zhu, J., & Yang, J. (2002, may). Subpixel eye gaze tracking. In *Automatic face and gesture recognition, 2002. proceedings. fifth ieee international conference on* (p. 124 -129).

DOI 10.16910/jemr.7.3.3

ISSN 1995-8692