

A Robust Approach to Automatic Iris Localization

ChengzheXu, Tauseef Ali, and Intaek Kim*

*Signal Processing Lab, Department of Communication Engineering,
Myongji University, Yongin 449-728, Korea*

(Received November 10, 2008 : revised February 2, 2009 : accepted February 17, 2009)

In this paper, a robust method is developed to locate the irises of both eyes. The method doesn't put any restrictions on the background. The method is based on the AdaBoost algorithm for face and eye candidate points detection. Candidate points are tuned such that two candidate points are exactly in the centers of the irises. Mean crossing function and convolution template are proposed to filter out candidate points and select the iris pair. The advantage of using this kind of hybrid method is that AdaBoost is robust to different illumination conditions and backgrounds. The tuning step improves the precision of iris localization while the convolution filter and mean crossing function reliably filter out candidate points and select the iris pair. The proposed structure is evaluated on three public databases, Bern, Yale and BioID. Extensive experimental results verified the robustness and accuracy of the proposed method. Using the Bern database, the performance of the proposed algorithm is also compared with some of the existing methods.

Keywords : AdaBoost, Candidate Point, Circular Template, Mean Crossing Function

OCIS codes : (070.5010) Pattern recognition and feature extraction; (100.2000) Digital image processing; (150.0150) Machine vision

I. INTRODUCTION

In the past two decades, face recognition has gained a lot of interest in the area of computer vision and pattern recognition. Because of the vast number of applications, topics such as face detection, face identification, face recognition, and facial expression analysis are attracting more and more attention. Automatic eye detection is one of the most important and fundamental problems to be solved among these research topics. Eyes are the most salient and stable feature in a human face, and many face detection algorithms [1-3] depend on robust eye detection. Face images are normalized using the position of eyes before comparing it with the database for face recognition. Furthermore, by observing a person's eyes, we can understand a person's desires, needs and emotions. Eye localization combined with eye tracking is crucial for facial expression analysis [4,5], human computer interaction, and attentive user interfaces.

II. PREVIOUS WORK

Several facts about eyes are utilized by different researchers to locate the position of eyes. Firstly eyes possess strong horizontal edges [6]. Secondly, the pupil appears darker than its surrounding. Some facial feature extraction algorithms search for local gray minima within the segmented face regions [7]. Zhang [8] located eyes using threshold binary images and using knowledge about facial features. However, such an algorithm is highly sensitive to the type of threshold methods and lighting conditions. In short, gray level information is helpful in finding several eye candidates. But using it alone may not be sufficient because of other facial features such as eyebrows, which also appear dark. Thirdly, eyes have their shape. An artificial template can be built according to the rough shape of the eye and eyebrow such that the correlation coefficient between the template and the eye image can be calculated [9]. Another way to represent the shape of the eye is by boundary edges, although the extraction of a feature boundary is not easy. Due to variation in face orientation or bad lighting conditions, the eye's boundary in an edge image would be

*Corresponding author: kit@mju.ac.kr

incomplete or broken. The deformable template based on the parameters of an eye model can extract the optimal eye boundaries once the template is initialized on the correct location of the eye [10,11]. Snake is another similar approach to the deformable template [12]. Given a good lighting condition, the edge image will have a circle around the iris because of the circular shape of iris. Hough transform can be used for the detection of eyes since it is suitable to detect shapes such as circles and ellipses.

Besides the above mentioned common approaches, recently some other eye detection techniques have been proposed which produce better results by combining several facts about eyes. Feng and Yuen [13] developed the variance projection function (VPF) to locate the position of eye by detecting the landmarks around the eye. The idea of VPF is extended to the generalized projection function (GPF) by Zhou and Geng [14]. They showed that the hybrid projection function (HPF), which is a special case of GPF, gives better results than VPF and integral projection function (IPF) for eye detection.

One of the best efforts was made by Kawaguchi and Rizon [7]. They located the iris using intensity and edge information. They combine feature template, separability filter, Hough transform and template matching to locate the iris. For face detection, Wu and Zhou [3] utilized size and intensity information to detect segments which are analogous to eyes from a gray scale image, and then used a special geometrical relationship to find possible eye pairs. Similarly, Han et al. [8] used morphological closing, conditional dilation and a labeling process to detect eye-analog segments. Hsu et al. [1] employed color information for eye detection.

Although much effort has been spent and some progress has been made, the problem of automatic eye detection is still not solved because of its complexity. Many factors, such as facial expression, face rotation in plane and depth, occlusion, lighting conditions, and so on, affect the performance of eye detection algorithms.

In this paper, we propose a robust algorithm for automatic iris localization. The algorithm detects the face in an input image using the AdaBoost algorithm. By changing the training data and increasing the false positive rate of the AdaBoost algorithm, we detect the candidate points for the eyes. The candidate points generated by AdaBoost are tuned such that two of the candidate points are exactly in the centers of irises. Mean crossing function and convolution template are used to select the irises of both eyes from the candidate points. Fig. 1 shows the steps in localizing iris centers of eyes.

III. FACE DETECTION

The task of eye localization is difficult since there are

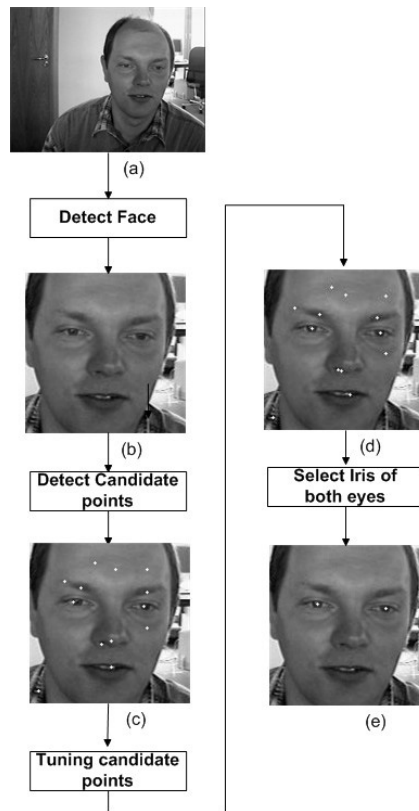


FIG. 1. TSteps of the proposed algorithm.

an almost infinite numbers of objects with eye-like features in an input image. On the other hand, face detection is easier, as face patterns have more features than eye patterns. We first detect the face in the input image. Then the problem is simplified due to the background being restricted to the face. It saves searching time and improves accuracy.

For face detection, Viola's method [15] is used. A robust face classifier is obtained by supervised AdaBoost learning. Given a sample set of training data $\{x_i, y_i\}$, the AdaBoost algorithm selects a set of weak classifiers $\{h_j(x)\}$ from a set of Haar-like rectangle features and combines them into a strong classifier. The strong classifier $g(x)$ is defined as follow:

$$g(x) = \begin{cases} 1 & \sum_{k=1}^{k_{max}} \alpha_k h_k(x) \geq \theta \\ 0 & otherwise \end{cases} \quad (1)$$

where θ is the threshold that is adjusted to meet the detection rate goal. The Haar-like rectangle features are easily computed using "integral image" representation. The "cascade" method quickly filters out non-face image areas. More details can be found in [15]. Training data is selected from different well-known face databases. All of the face images are first processed using gray-scale and size normalization to 20*20 pixels. Negative samples

are obtained using a bootstrapping method.

IV. EYE CANDIDATES DETECTION

By changing the training data and increasing the false-positive rate of the algorithm in section III, we build an eye candidate detector. The training data of [16] is used to detect several eye candidate points in the face region. A total of 7000 eye samples are used with the eye center being the center of the image and resized to 16*8 pixels. Because in this step the face region is already detected, the negative samples are taken only from the face images. We set a low threshold and accept more false positives. On the average, we get 15 eye candidates out of the detector.

V. TUNING CANDIDATE POINTS

The candidate points generated by AdaBoost, contain two points which represent eyes. But in most cases AdaBoost can't point out the centers of irises. So we need to shift the candidate points within a small size of neighborhood so that two of the candidate points are exactly in the centers of irises. The separability filter proposed by Fukui and Yamaguchi [17] is modified in an efficient way to shift the candidate points within a small sized neighborhood. If we form two concentric circles around the center of the iris and measure the separability between the small circle and the outer part of the large circle as shown in the template of Fig. 2, the separability value is greater for irises as compared to other candidates. For a neighborhood of size $m \times n$, we have $m \times n$ new candidate points for each of original candidate points. By using the template in Fig. 2, the separability value (η) is computed for each point in the neighborhood by the following equation.

$$\eta = \frac{V_B}{V_W}$$

$$V_B = n_1(\bar{p}_1 - \bar{p}_m)^2 + n_2(\bar{p}_2 - \bar{p}_m)^2 \quad (2)$$

$$V_W = \sum_{i=1}^{n_1} [I_1(x_i, y_i) - \bar{p}_1]^2 + \sum_{i=1}^{n_2} [I_2(x_i, y_i) - \bar{p}_2]^2$$

Where η is the separability value between region R_1 and R_2 . V_W and V_B are the within-class scatter value and the between-class scatter value, respectively, n_k ($k = 1; 2$) is the number of pixels in R_k ; p_k ($k = 1; 2$) is the average intensity in R_k ; P_m is the average intensity in the union of R_1 and R_2 , and $I_k(x_i, y_i)$ is the intensity values of pixels (x_i, y_i) in R_k .

The separability value can't determine which region is the dark one between R_1 and R_2 , so we add the condition that if $p_1 > p_2$, discard candidate point. It means that if the R_1 region is brighter than the R_2 region, we cancel out those candidate points.

Using equation 2 and the circular template shown in Fig. 2, we repeat the process of forming a neighborhood and selecting a new candidate for each original candidate point. Separability values for each of the points in the neighborhood are determined by varying the radius in a range $\{R_L, \dots, R_U\}$. In our experiments we used $R_L = 3$ and $R_U = 6$. The point in the neighborhood which gives maximum separability is considered as the new candidate point. In our experiment we checked for several sizes of neighborhoods, and we have concluded that a 9*9 neighborhood is most efficient and correct. We also find the separability values for each new candidate point and its corresponding optimal radius R among $\{R_L, \dots, R_U\}$. These separability values and radius for new candidate points are used later. Fig 1(d) shows the new candidate points.

VI. SELECTING IRIS PAIR

We combine three metrics to measure the fitness of each candidate point with the iris and to select the iris pair. The first metric is based on the separability value determined in the previous section. The other two metrics are based on the mean crossing function and the convolution template proposed in this section.

Mean crossing function

The eye has the unique property in the face in that it possesses very low gray levels inside the iris and very high gray levels to its left and right (sclera of eye). A simple two-dimensional mean crossing function which utilizes this fact about the eye is used to measure the fitness of each eye candidate. A rectangular subregion is formed around each iris candidate. The size of the subregion is depicted in Fig. 3, where R is the radius of the candidate determined in section V. The values

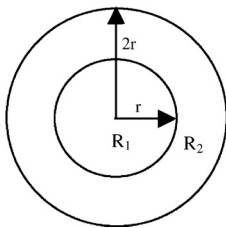


FIG. 2. Circular template (R_1 is the inside region of the smaller circle and R_2 is the region between the two concentric circles).

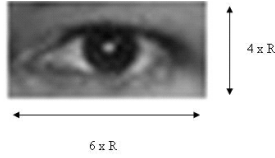


FIG. 3. Subregion for mean crossing function.

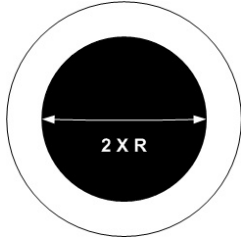


FIG. 4. Convolution template.

6 and 4 are intuitively chosen to cover the whole eye region including the iris and the white of the eye.

A subregion of the form shown in Fig. 3 is formed around each candidate point. Now the mean value is determined for the subregion as

$$\mu = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N I(i, j) \quad (3)$$

where M , N is the length and width of the subregion and $I(i, j)$ is the intensity of pixel (i, j) . The subregion is scanned horizontally and the mean crossing function [18] for pixel (i, j) is computed as follows:

$$\mu C(i, j) = \begin{cases} 1 & ; \text{If } (I(i, j) \geq \mu + A \text{ and } I(i, j+1) \leq \mu - A) \\ 1 & ; \text{If } (I(i, j) \leq \mu - A \text{ and } I(i, j+1) \geq \mu + A) \\ 0 & ; \text{otherwise} \end{cases} \quad (4)$$

where A is a constant. We verified by our experiments that $A = 10$ gives best results for a range of illumination conditions. The horizontal mean crossing value for the subregion is determined as

$$\mu C_{\text{subregion}} = \sum_{i=1}^M \sum_{j=1}^N \mu C(i, j) \quad (5)$$

In a similar way, the vertical mean crossing function is evaluated by scanning vertically the subregion.

To find the final mean crossing value for the subregion, we linearly add both mean crossing numbers.

Convolution with edge image subregion

First we find the edge image of the subregion around the candidate point. The size of the subregion is the same as the mask in Fig. 4. The subregion is convolved with the convolution kernel shown in Fig. 4 with the edge image of the subregion. In order that the convolution gives the best result we need thin edges. Therefore we

used a Canny edge detector although it requires more time than a Sobel edge detector. The radius value of each candidate is determined in section V. The center of the template is placed on the candidate point and the value of convolution is computed.

The process is repeated for each candidate. The resultant signal from convolution is summed up and a single value is obtained. The greater is the value from the convolution, the more is the probability that the candidate point is an iris.

Now, we define fitness of an eye candidate by the following equation.

$$\text{fitness}(C_x) = \frac{\mu C_{\text{subRegion}}(C_x)}{\sum_{i=1}^N \mu C_{\text{subRegion}}(C_i)} + \frac{\text{Conv}(C_x)}{\sum_{i=1}^N \text{Conv}(C_i)} + \frac{\text{Seperability}(C_x)}{\sum_{i=1}^N \text{Seperability}(C_i)} \quad (6)$$

Where N is the total number of candidate points, $\mu C_{\text{subRegion}}(C_j)$, $\text{Conv}(C_j)$, are the mean crossing value and convolution result for candidate C_j and $\text{Seperability}(C_j)$ is the separability value for candidate C_j computed in section V. To select an iris pair, first we short-list the pairs using the following geometric conditions.

(a) The horizontal orientation of the candidate points in a pair must be within the range of -40° to 40° .

(b) If l is the total horizontal width of the face region, then the horizontal distance between candidate points in a pair must be greater than $l/4$.

Now, candidate pair with the maximum fitness is taken as the iris pair according to the following equation.

$$\text{fitness}(\text{Pair}(C_x, C_y)) = \text{fitness}(C_x) + \text{fitness}(C_y) \quad (7)$$

VII. EXPERIMENTAL RESULTS

In this section the experimental results obtained on three public databases, Bern [19], BioID [20] and Yale A [21] are shown. Two criteria are used for the evaluation of the performance of an eye detection algorithm. First is the detection rate. It refers to the ratio of the number of images for which two eyes are correctly detected to the total number of images tested. Second is the localization accuracy. It refers to the disparity between the manually detected eye position and the automatically detected eye position. Usually the larger of the two eye disparities in a face image is adopted as the accuracy measure of eye detection. Jesooriky et al. [22] proposed the relative error to judge the quality of eye detection, which is defined by

TABLE 1. Comparison of correct eye detection rate of proposed method with Kawaguchi and Rizon's, template matching and eigenspace method using 150 Bern images.

Algorithm	Detection Rate (%)
Kawaguchi and Rizon [7]	95.3%
Template matching [7]	77.9%
Eigenface method using 50 training samples[7]	90.7%
Eigenface method using 100 training samples[7]	93.3%
Proposed method	99.33%

$$d_{eye} = \frac{\max\left(\|C_l - \tilde{C}_l\|, \|C_r - \tilde{C}_r\|\right)}{\|C_l - C_r\|} \quad (8)$$

where C_l , and \tilde{C}_l are the manual and automatic eye centers, respectively. Obviously this metric is not dependent upon image resolution. There is a general agreement [3, 4, 22] that $d_{eye} < 0.25$ is considered correct eye localization (to claim eye detection). This precision roughly corresponds to a distance smaller than the eye width. However, this accuracy level may not be sufficient when the localized positions are used for the initialization of subsequent techniques such as face recognition.

In our experiments, we use the criteria for correct iris localization as $d_{eye} < 0.125$ also adopted by [23]. Because the radius of an iris, denoted by R , is about 1/4 of an eye width, our criteria correspond to

$$\max\left(\|C_l - \tilde{C}_l\|, \|C_r - \tilde{C}_r\|\right) < R \quad (9)$$

The criteria in eq. 9 also mean that if both the left and right eye positions detected hit irises, the eye detection is considered a success. The standard is very close to Kawaguchi and Rizon's[7].

Experiments and comparison using Bern images

Bern 150 images [18] are widely used for comparison purposes. We tested our algorithm on this database for comparing the performance of our proposed algorithm with previous work. In the next section we have shown results on Yale A and BioID databases. Generally the 150 Bern images used in this section are simple as compared to Yale A and BioID databases. These images don't contain closed eye subjects or subject wearing glasses, and they are taken under good lighting conditions.

Table 1 shows the performance comparison of our proposed eye detection method with Kawaguchi and Rizon's [7], the template matching and the eigenface method using 150 Bern images.

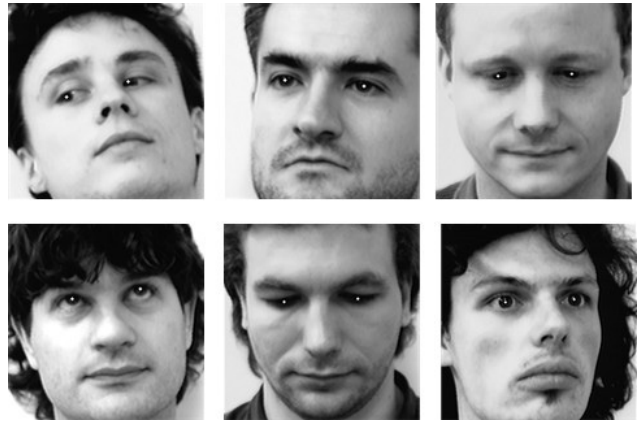


FIG. 5. Some of the successful iris localization results from Bern database.



FIG. 6. The only image from 150 Bern images for which algorithm failed to localize iris pair correctly.

TABLE 2. Results of the proposed algorithm on Yale A and BioID database.

Error Category	Yale A	BioID
Error on closed eyes images	3.03%	2.00%
Error on partially closed eyes images	1.20%	1.33%
Error on open eyes images	3.64%	5.33%
Total Error	7.88%	8.67%

In Fig. 5 we have shown some of the Bern face images for which the proposed algorithm can correctly locate the iris of both eyes.

FIG. 6 shows the single face image out of 150 for which the algorithm can't correctly detect the iris pair.

Experiments with Yale A and BioID Databases

The Yale A database contains 165 grayscale images of 15 individuals [20]. There are 11 images per subject, one per different facial expression or configuration: center-light, wearing glasses, happy, left-light, wearing no glasses, normal, right-light, sad, sleepy, surprised, and wink. The database also contains closed eyes images and images in which subjects are wearing glasses. Images also vary in lighting conditions, ethnicity, and pose. The algorithm is tested on all 165 data images. In 152 images out of 165, the algorithm can correctly locate the position of the iris according to eq. 9. The overall success rate of the algorithm using Yale A database is 92.12%

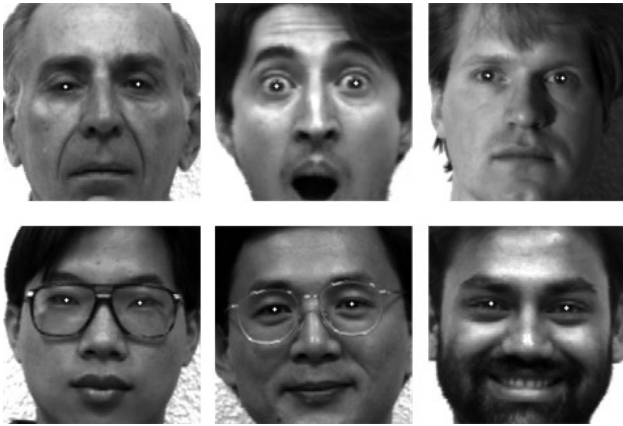


FIG. 7. Some examples of successful iris localization from Yale A database.



FIG. 8. Some examples of successful iris localization from BioID database.

including images having closed eye subjects and subjects wearing glasses.

The BioID dataset [19] consists of 1521 gray level images. Each one shows the frontal view of a face of one out of 23 different test persons. Like Yale A, this database also contains closed eye images and images in which subjects are wearing glasses. The algorithm is tested on 300 data images. In 274 images out of 300, the algorithm can correctly locate the position of eyes according to eq. 9. The overall success rate of the algorithm is 91.33% including closed eye images and partially closed eye images.

In Table 2, we have tabulated the result of the method when tested on these two databases. To show the performance of the algorithm in more detail, we have classified the error into three classes: Error caused by closed eyes, partially closed eyes, and normal data (subjects with open eyes including subject wearing glasses). The performance of the algorithm is low on Yale A and BioID databases as compared to Bern because of the bad lighting conditions, subject with partially or fully closed eyes, sub-



FIG. 9. Examples of incorrect iris localization from Yale A Database.



FIG. 10. Examples of incorrect iris localization from BioID Database.

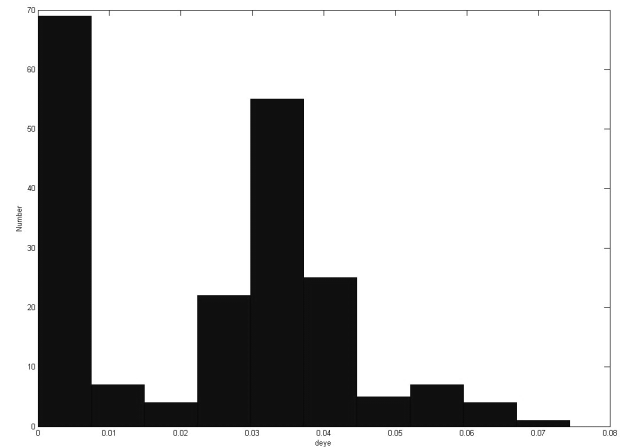


FIG. 11. Iris localization accuracy obtained from 200 test images.

jects wearing glasses and bright reflection on eyes.

Because the system utilizes the mean crossing function and the convolution with circular template, if the eyes are closed then it is not possible to get a circle of edge points and there is less intensity variation in the eye region. Also bright reflection around the eye eliminates the distinguishing feature of the eye utilized by our method of identification.

FIG. 7 and FIG. 8 show some of the successful results from the Yale A and BioID databases respectively.

FIG. 9 and FIG. 10 show some of the unsuccessful results from the Yale A and BioID databases respectively. In Fig. 11 the accuracy of iris localization is plotted for 200 test images using eq. 8.

VIII. CONCLUSION

Our approach can robustly localize the iris from frontal view face images as well as those looking to left, right, downward and upward. We achieve iris localization in four steps: (1) Face Detection (2) Eye candidates detection (3) Tuning candidate points and (4) iris selection. The contribution of this paper mainly starts from step 3, after eye candidate points are found by AdaBoost. In step 3 candidate points are shifted in a way that greatly improves the accuracy of iris localization. Step 4 utilizes three metrics and can robustly filter out candidate points. For testing purposes, three popular databases, Bern, Yale and BioID are used. Tested data include different lighting conditions, ethnicity, pose and facial expressions such as neutral, smile, anger, wink, drowsiness, etc. In our approach, we improve the accuracy of eye localization by tuning the candidate points generated by AdaBoost. The effectiveness of the algorithm is thoroughly analyzed both quantitatively and qualitatively by measuring the error rate and localization accuracy respectively. We achieve a detection rate of 99.33% on Bern images and compare it with state-of-the-art works available. An overall detection rate of 92.12% and 91.33% are achieved on Yale A and BioID databases respectively.

REFERENCES

1. R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 696-706 (2002).
2. C. C. Han, H. Y. M. Liao, G. J. Yu, and L. H. Chen, "Fast face detection via morphology-based pre-processing," *Pattern Recognition* **33**, 1701-1712 (2000).
3. J. Wu and Z. H. Zhou, "Efficient face candidates selector for face detection," *Pattern Recognition* **36**, 1175-1186 (2003).
4. B. Fasel and J. Luetten, "Automatic facial expression analysis: a survey," *Pattern Recognition* **36**, 259-275 (2003).
5. Y. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 97-115 (2001).
6. D. Maio and D. Maltoni, "Real-time face location on grayscale static images," *Pattern Recognition* **33**, 1525-1539 (2000).
7. T. Kawaguchi and M. Rizon, "Iris detection using intensity and edge information," *Pattern Recognition* **36**, 549-562 (2003).
8. L. Zhang and P. Lenders, "Knowledge-based eye detection for human face recognition," in *Proc. the 4th Int. Conf. on Knowledge-Bases Intelligent Systems & Allied Technologies* (Brighton, UK, 2000), vol. 1, pp. 117-120.
9. M. Betke and W. J. Mullally, "Preliminary investigation of real-time monitoring of a driver in city traffic," in *Proc. IEEE Intell. Vehicles Symposium* (Dearborn, MI, USA, 2000), pp. 563-568.
10. A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *Int. Journal of Computer Vision* **8**, 99-111 (1992).
11. Y. Tian, T. Kanade, and J. F. Cohn, "Dual-state parametric eye tracking," in *Proc. Int. Conf. on Face and Gesture Recognition* (Grenoble, France, 2000), pp. 110-115.
12. M. Pardas, "Extraction and tracking of the eyelids," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* (Istanbul, Turkey, 2000), pp. 2357-2360.
13. G. C. Feng and P. C. Yuen, "Variance projection function and its application to eye detection for human face recognition," *Pattern Recognition Lett.* **19**, 899-906 (1998).
14. Z. H. Zhou and X. Geng, "Projection functions for eye detection," *Pattern Recognition* **37**, 1049-1056 (2004).
15. P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. Computer Vision and Pattern Recognition Conference 2001* (Hawaii, USA, 2001), vol. 1, pp. 511-518.
16. M. Castrillon-Santana, J. Lorenzo-Navarro, O. Djeniz-Suarez, J. Isern-Gonzalez, and A. Falcjon-Martel, "Multiple face detection at different resolutions for perceptual user interfaces," in *Proc. 2nd Iberian Conference on Pattern Recognition and Image Analysis 2005*, J.S. Marques et al. Ed., LNCS, (Estoril, Portugal, 2005), vol. 3522, pp. 445-452.
17. K. Fukui and O. Yamaguchi, "Facial feature point extraction method based on combination of shape extraction and pattern matching," *Systems and Computers in Japan* **29**, 49-58 (1998).
18. C.-H. Lin and J.-Ling, "Automatic facial feature extraction by genetic algorithms," *IEEE Trans. Image Process* **8**, 1057-1149 (1999).
19. <http://iamwww.unibe.ch/~kiwww/staff/achermann.html>
20. <http://www.bioid.com/downloads/facedb/index.php>
21. <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>
22. O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *Proc. the Third International Conference on Audio- and Video-based Biometric Person Authentication 2001*, (Halmstad, Sweden, 2001), pp. 90-95.
23. J. Song, Z. Chi, and J. Liu, "A robust eye detection method using combined binary edge and intensity information," *Pattern Recognition* **39**, 1110-1125 (2006).