International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 24, No. 3 (2010) 475–498

DOI: 10.1142/S0218001410008020

A NOVEL METHOD FOR HORIZONTAL EYE LINE DETECTION UNDER VARIOUS ENVIRONMENTS

MIN-QUAN JING* and LING-HWEI CHEN†

Department of Computer Science
National Chiao Tung University, 1001 Ta Hsueh Rd.
Hsinchu, Taiwan 30050, R.O.C.
**ching@debut.cis.nctu.edu.tw*
†*lhchen@cc.nctu.edu.tw*

The eye line is defined to be a horizontal line passing the two eyes of a human face. It can be used to help locate the true positions of eyes and face. In this paper, we propose a method to extract the horizontal eye line of a face under various environments. Based on the facts that the eye color is very different from skin color and the gray level variance of an eye is high, some eye-like regions are first located. Then the horizontal eye line is extracted based on the located eye-like regions and some geometric properties in a face. Experimental results are given to show that the proposed method is robust under a wide range of lighting conditions, different poses and races. The detection rate for HHI face database is 95.63%. For Champion face database, the detection rate is 94.27%.

Keywords: Face detection; various lighting environments; eye line; skin color.

1. Introduction

In a drowsy driving warning system, the driver's mental condition can be revealed by measuring the eye movement from the eye location.[2] How to locate the eyes of a driver under extreme lighting condition is an important issue for a successful intelligent transportation system. Detecting the eye line can help locate eyes. Moreover, the detected eye line can be used to do face detection. Although face detection has been studied for more than 20 years, developing a human face detector under various environments is still a challenging task. Some factors make face location difficult. One is the variety of colored lighting sources, another is that facial features such as the eyes may be partially or wholly occluded by shadow generated by a bias lighting direction; others are races and different face poses with/without glasses. In order to overcome this problem, Hjelmas and Low[3] have presented an extensive survey of feature-based and image-based algorithms for face detection. In feature-based methods, the most widely used technique is skin color detection based

on one of the color models. Most image-based approaches are based on multi-resolution window scanning to detect faces at all scales, which make them computationally expensive. Also, Yang[21] classified the face detection approaches into four categories: knowledge-based, feature invariant, template matching, and appearance-based, addressing one of the difficulties of face detection as lighting source variation. Several other methods[1,4−20] were also proposed. Some[9,17,19] use eigenface, neural network and support vector machine (SVM) to detect faces of restricted poses under normal lighting condition. Chow[1] proposed an approach to locate facial features (such as eyes, eyebrows) by using an eigenmask under different lighting conditions. However, the method failed to locate the facial features for some kinds of poses (such as near-profile and profile face). Shin and Chuang[13] used shape and contour information to locate the face in a plain background and normal lighting condition. However, it is difficult to detect the predefined shape of a face in a complex background and under various lighting conditions. Wu *et al.*[20] proposed an algorithm to locate face candidates using eye-analog segment information. The algorithm would fail when a person wears glasses. Some methods[6,8,15] used mosaic, edge and mathematical morphology to detect eyes, however, these methods failed to locate the eyes when a face is under a poor lighting condition (such as bias-light source). Wang[18] proposed a boosted based algorithm to detect a face. But the training phase of the method is time consuming and it is only designed for the frontal face. Shih[12] proposed a color based method showing reasonable performance in terms of detection rate. However, the detection failed under extreme lighting condition.

As mentioned above, extreme lighting conditions (such as colored lighting source and bias lighting direction), different subject poses, glasses, races and complex background are factors that make face detection difficult. To solve the problem caused by these factors, we propose a method to extract the horizontal eye line of a face and some eye-like regions under various environments. The extracted eye line and eye-like regions can be further used to help extract the true positions of eyes and face.

The proposed method contains several steps. First, we used skin colors to extract candidate skin regions. Next, an eye-like region detector based on intensity and color information was provided to explore all possible eye-like regions in each candidate skin region. A lighting distribution based algorithm was then be presented to remove some false eye regions. Finally, based on the extracted eye-like regions, the horizontal eye line for a candidate skin region can be located using the gray level distribution of the skin region.

The remainder of the paper is described as follows. In Sec. 2, a color-based detector is presented to extract some candidate skin regions. In Sec. 3, a method is proposed to explore eye-like regions in each candidate skin region. Section 4 will present an algorithm to extract the eye line of a face. In Sec. 5, some experimental results based on HHI and Champion face databases will be given to demonstrate the effectiveness of the proposed method. Section 6 comprises the conclusion.

2. The Proposed Skin Region Detector

In several face detection systems,[5] skin color plays a major role for segmenting facial regions from an image. There are several color models utilized to label pixels as skin, these include RGB, HSV and YCrCb color spaces.[4,10,11] Since a face image is usually taken under unpredicted lighting conditions, RGB color-model is sensitive to light and unsuitable for representing skin color. In YCbCr model, Cr represents the red degree.[5] Since human skin color tends to red, we will adopt the YCbCr color model. The transformation between YCbCr model and RGB model is shown in Eq. (1).

$$\begin{cases} Y = 0.257 \times R + 0.504 \times G + 0.098 \times B + 16 \\ C_b = -0.148 \times R - 0.291 \times G + 0.439 \times B + 128 \\ C_r = 0.430 \times R - 0.368 \times G - 0.071 \times B + 12.8 \end{cases} \quad (1)$$

2.1. *Skin region extraction*

As mentioned above, human skin color tends to red despite the race, according to Eq. (1), red color will have higher Cr value. In addition, since high lighting will raise the R, G and B values and make these three values closer, according to Eq. (1), this will reduce the Cr value. Based on this fact, Cr value will be used to determine whether a pixel is skin. Two different thresholds Cr_{th1} (142) and Cr_{th2} (132) are adopted to bound skin color. For an input image, we first use Cr_{th1} as threshold to obtain the candidate's skin regions. Then, for the same input image, we use Cr_{th2} as threshold to obtain other skin regions. Figure 1 shows some results of applying the skin region extractor on different racial faces. From this figure, we can see that whatever the light-conditions (and the human races change), most skin pixels are extracted (white pixels). After skin pixels are extracted, those connected skin pixels are grouped to form a set of candidate skin regions, each region is bounded by a red rectangle [see Figs. 1(b) and 1(c)]. From Fig. 1, we can see that some extracted candidate skin regions are not true skin. In the next section, we will present a filter to remove some false skin regions.

2.2. *Skin region filter*

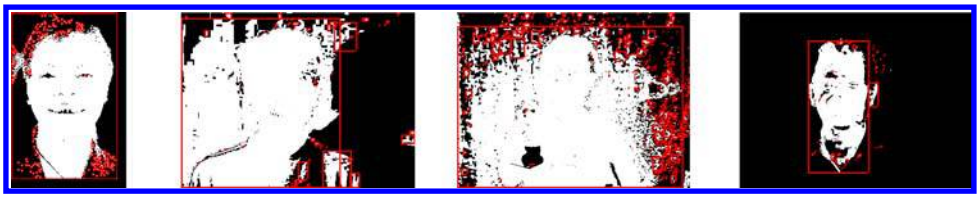
The skin region filter is designed to remove false skin regions based on four features: size, shape, uniform-content and geometric relation. Each feature can be used to remove several false regions. The detail is described as follows.

(1) **Size:** A human face should have a reasonable width and height, thus, a candidate skin region will be removed if the width or height of its bounding rectangle is less than a threshold S_{th}. In this paper, we set S_{th} to be 15 [see Figs. 2(b) and 2(d)]. On the other hand, if the width of its bounding rectangle is larger than a half of the input image width, the region will also be removed.

(2) **Shape:** In general, the shape of a face should be like an ellipse. Thus, if a region looks like a horizontal or vertical thin bar, it should be removed [see Fig. 2(b)].

(a) Four images taken under normal, bias colored and high lighting environments.

(b) The extracted candidate skin regions using threshold 142.



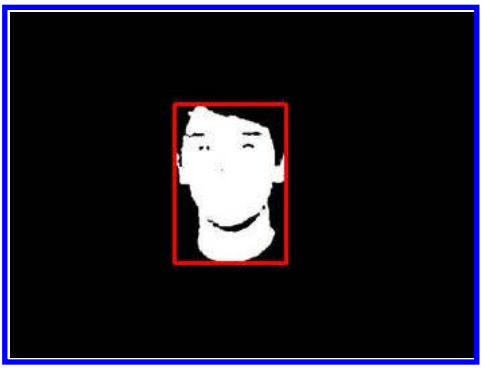
(c) The extracted candidate skin regions using threshold 132.

Fig. 1. Some results of applying skin region extractor on different races under various lighting environments.

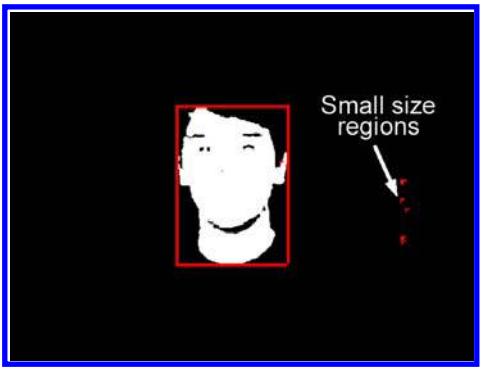
Here, if a region has $H_s/W_s \geq V_{th}$, it is considered as a vertical thin bar. And if a region has $W_s/H_s \leq H_{th}$, it is considered as a horizontal thin bar. Note that H_s and W_s are the height and the width of a region, and in the paper, we set V_{th} to be 2.5 and H_{th} to be 0.5.

(a) A face image. (b) Candidate skin regions using 132 as threshold.

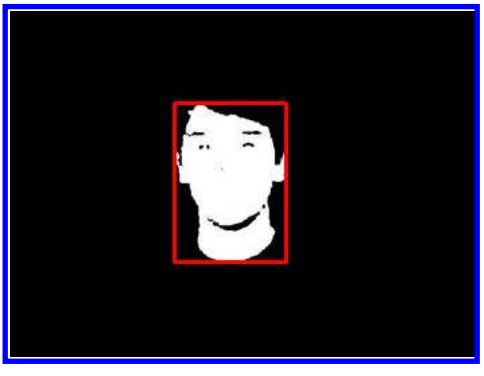
Fig. 2. Some examples for illustrating the skin region filter.



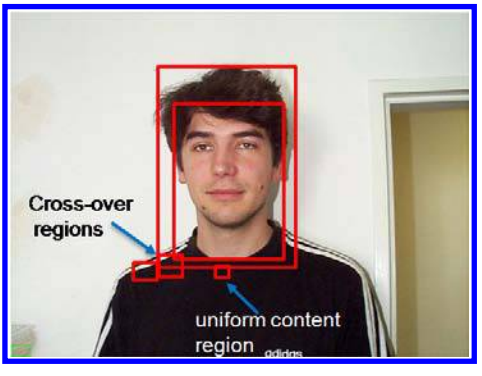
(c) The result of removing regions of small sizes and improper shapes in (b).



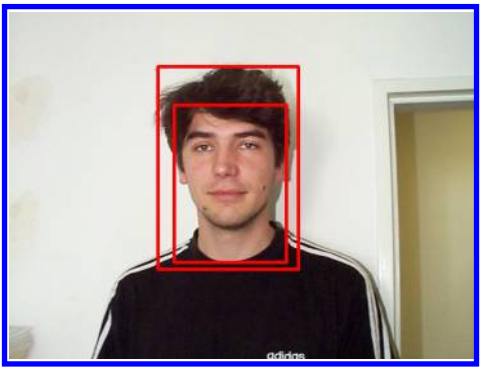
(d) Candidate skin regions using 142 as threshold.



(e) The result of removing small regions in (d).



(f) The result of merging (c) and (e), and imposed on (a).



(g) The result of removing uniform content regions and cross over regions in (f).

Fig. 2. (*Continued*)

(3) **Uniform-Content:** A uniform content region should not be a face since a face contains eyes and eyebrows. Thus, if the standard derivation of the gray values in the bounding rectangle of a candidate skin region is less than a predefined threshold U_{th} (10), the region will be removed [see Fig. 2(f)].

(4) **Geometric relation:** If two candidate skin regions crossover and the small one is less than a half of the big one, then the small one is removed [see Fig. 2(g)]. On the other hand, if a small region is totally covered by another big region, then both regions are preserved [see Fig. 2(g)]. For this case, in the later process, the small one is first used to detect the eye line. If an eye line is found, the detection for the big one will be skipped.

3. The Proposed Eye-Like Region Detector

In this section, we will provide a method to extract eye-like regions from those remaining candidate skin regions. Figure 3 shows the block diagram of the proposed eye-like region detector. First, based on intensity and skin color information, some eye-like regions are extracted. Next, some false regions are removed. Since the true eye may not be located in the center of the eye-like region, a refining algorithm is presented to adjust the eye-like region. Finally, some false eye-like regions appearing in hair or shadow part will be removed, and isolated eye-like regions are also removed.

In Ref. 1, luminance contrast and eye shape are used to detect an eye. However, under various lighting environments, unpredicted shadow appearing on face makes eye detection difficult. Figure 4 shows some face images taken under various lighting environments. To treat this problem, we use two fundamental eye properties to extract eye-like regions. The first property is that the pupil and iris' gray values are lower than the gray values of skin. Based on this property, an intensity based technique is provided to locate eye-like regions in the bounding rectangle of each candidate skin region. The second property is that under noncolored lighting source condition, the eye color will be very different from skin color. Based on this property, we consider those small nonskin color regions in the bounding rectangle of a candidate skin region as eye-like regions.

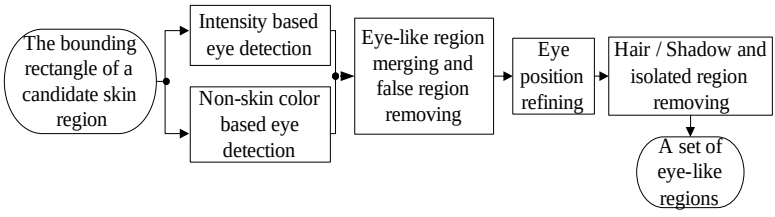


Fig. 3. The block diagram of the eye-like region detector.

(a) The normal lighting condition. (b) Top lighting source.

(c) Colored and left bias lighting source. (d) High lighting source.

Fig. 4. Subjects taken under various lighting enviornments.

3.1. *Intensity based eye-like region extraction*

After extracting candidate skin regions, an intensity based detector will be first provided to extract eye-like regions from the bounding rectangle of each candidate skin region. Figure 5 shows its flowchart.

Based on the fact that the pupil's gray values are lower than skin gray values, the detector adopts a bilevel thresholding to get the eye-like regions. First, for a candidate skin region, several different threshold values, $t1$, are used for bilevel thresholding, the maximum and minimum of $t1$ are named as t1_MAX and t1_MIN,

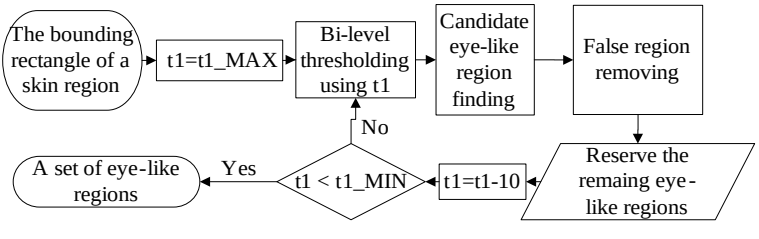


Fig. 5. The flowchart of the intensity based eye-like region detector.

respectively. In fact, a face area occupies at least 20% of a candidate skin region. In order to treat various lighting conditions, t1_MAX is set as the average gray value of 20% pixels with maximum gray values in the candidate skin region, and t1_MIN is set to be the average gray value of 5% pixels with minimum gray values in the candidate skin region. At the beginning, a bilevel threshold operator with t1_MAX as the initial threshold is applied to extract eye-like regions, and then the operator is applied iterately with $t1$ reduced by a value I_{th} (here, ten is used) each time until $t1$ reaches t1_MIN. Figure 6 shows an example for evaluating t1_MAX and t1_MIN, and the result of applying the intensity based eye detector.

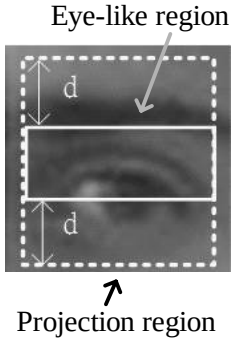
After applying a bilevel thresholding operator each time, each resulting black region is considered as a candidate eye-like region. However, some black regions, which may be hair region or small noise caused by uneven lighting, are false ones. Here, we provide three filters to remove these false eye-like regions. One is geometric filter, which is used to remove too small or too large regions. A black region with size less than S_{th2} (here, S_{th2} is set as 5×5) or its width (height) larger than S_{th3} (here, we set S_{th3} to be $1/4$ skin region width) will be removed. One is a statistical filter, if the standard derivation of the gray values of an eye-like region is lower than a predefined threshold, the region is considered as a false eye-like region and then removed. Here the threshold we chose is ten. From the structure of an eye, we can see that the iris and pupil's gray levels are lower than skin. Based on this property, the third filter called projection filter is provided to remove each black region with its average gray value larger than that of the upper or lower neighboring areas. A horizontal gray value projection histogram (see Fig. 7) is used to implement the filter. If the histogram value of the middle position is larger than the histogram value of the highest peak above or below the position, we removed the region. Figure 8 shows the final result of applying the intensity based eye-like region detector to a face image, each extracted eye-like region is enclosed by a red rectangle.

3.2. *Nonskin color based eye detector*

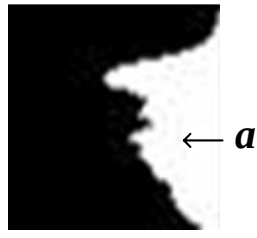
Under a very low lighting environment, an eye-like region extracted by intensity based detector may contain a large noneye part [see the left eye in Fig. 8(b)]. This

(a) A face region. (b) The histogram of detected gray pixels within the skin region. (c) Bilevel thresholding results of (a) using different thresholds.

Fig. 6. An example of the bilevel thresholding results.

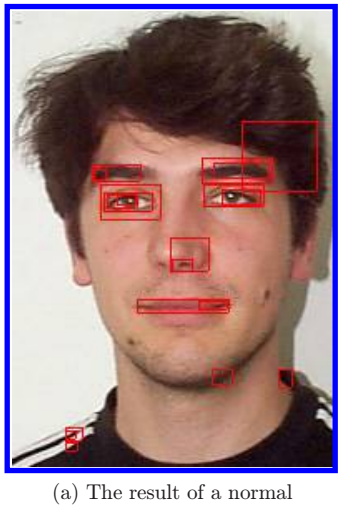


(a) The projection region extended from eye-like region.

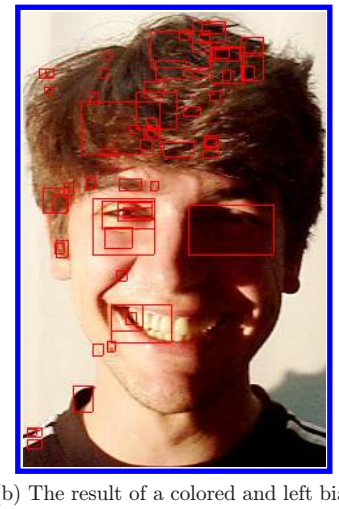


(b) The projected histogram of the projection region in (a).

Fig. 7. An example of the horizontal gray value projection for an eye.



(a) The result of a normal lighting source face.



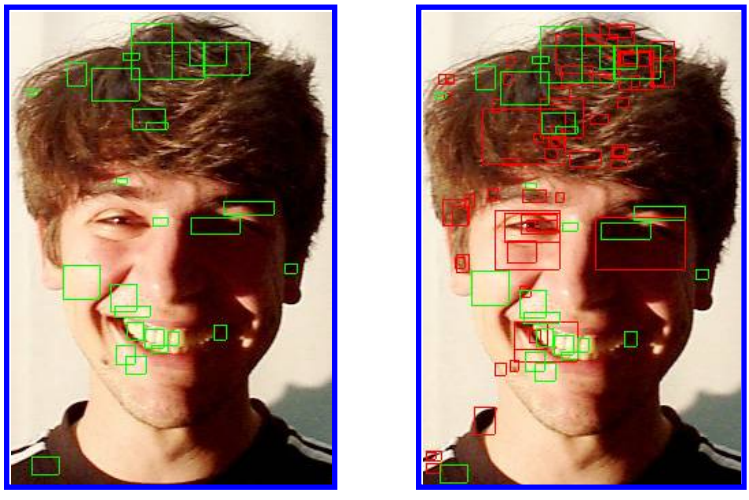
(b) The result of a colored and left bias lighting source face.

Fig. 8. The results of applying the intensity based algorithm.

will make the later process hard. To treat this problem, we use nonskin color information to locate the eyes. First, nonskin color regions within a candidate skin region are considered as eye-like regions. The three false eye-like region filters described in Sec. 3.1 are then used to remove some false regions. Figure 9(a) shows the result of applying the nonskin color based eye detector. Each eye-like region is enclosed by a green rectangle. From this figure, we can see that the left eye was precisely located. Figure 9(b) shows the combined result of Figs. 8(b) and 9(a).

3.3. *False eye-like region removal*

The obtained eye-like regions still contain some false regions; we can classify these regions into five classes: overlapped, hair reflecting, beard/clothes, isolated and



(a) The eye-like regions located based on nonskin color information.

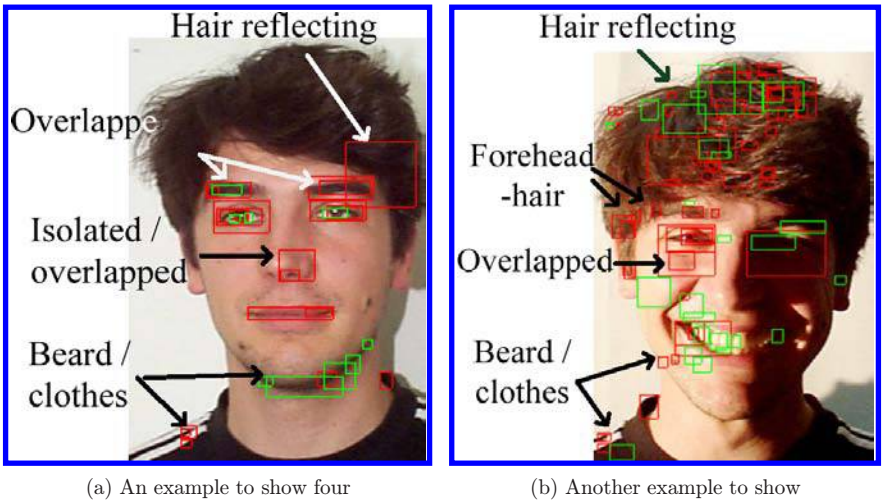
(b) The combined result of Figs. 8(b) and 9(a).

Fig. 9. The result of applying the proposed intensity and nonskin color based eye detectors.

forehead-hair (see Fig. 10). In the following, we will provide some procedures to remove these regions.

3.3.1. *Overlapping/hair reflecting/beard/clothes false region removal*

For the overlapping class, if one eye-like region totally covers a small eye-like region, the covering region is removed. To remove hair reflection regions, first, a horizontal gray level projection histogram [see Fig. 11(b)] is created. Then, the first peak



(a) An example to show four classes of false eye-like regions.

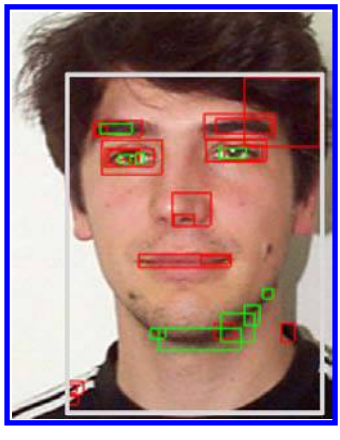
(b) Another example to show forehead-hair class.

Fig. 10. Five false region classes.

location from the top of the projection histogram is defined as an upper line which usually indicates the forehead. Each eye-like region intersecting the upper line is removed. For the beard/clothes class, we first define a bottom line with distance $h/2$ from the upper line, where h is the distance between the upper line and the bottom of the skin region. Then all eye-like regions below the line are removed. Figure 11(d) shows the result of applying the removing procedure to Fig. 11(a).

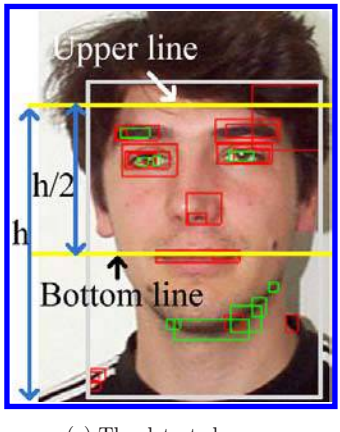
3.3.2. *Eye position refining and isolated false region removal*

For a nonrotated human face, an eye pair should be located at a near-horizontal line. Thus for an eye-like region, if we cannot find another region at its left or right, then this region is called an isolated region and should be removed. However, under

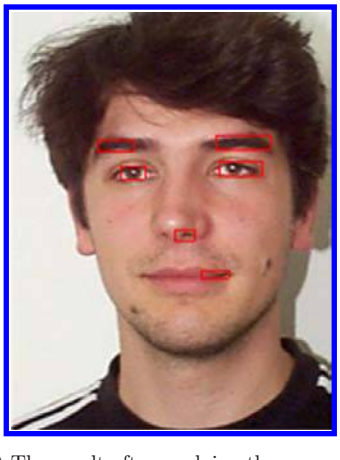


(a) A bounding rectangle of a candidate skin region.

(b) The horizontal projection histogram of the bounding rectangle in (a).

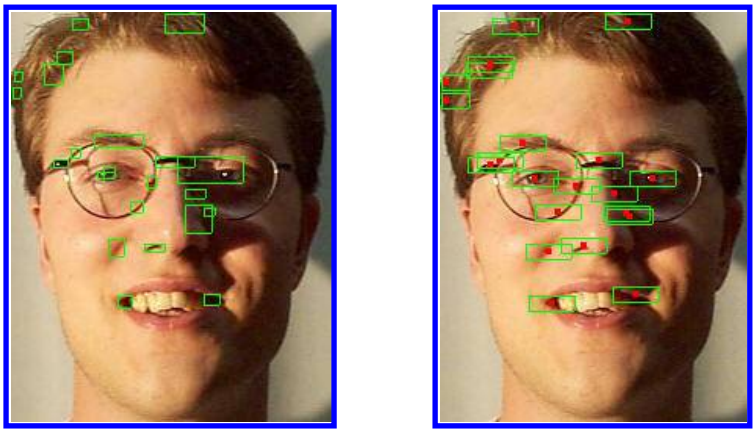


(c) The detected upper and bottom lines.



(d) The result after applying the removing procedure presented in Sec. 3.3.1 to (a).

Fig. 11. The result of applying the false eye-like removing procedure.



(a) An example to show an eye not located at the center of the eye-like region.

(b) The result of applying the eye position refining procedure to (a).

Fig. 12. The result of applying the eye position refining procedure.

various environments, a true eye may not be located at the center of the extracted eye-like region [see Fig. 12(a)], we are then unable to know where the eye is. In order to treat this problem, we provide an eye position refining method to locate the true eye position in the eye-like region. First, a set of eye templates is created, then these templates are used to refine the eye positions. For template creation, first, we classify the frontal images of HHI into three classes according to the lighting conditions: normal, high light and low light. Next, one image is selected randomly from each class. Considering the effect of glasses, the normal class has two images selected with/without wearing glasses. For each image, three templates are extracted (see Table 1). An eye area including eyelid is taken as the first template. From the first template, an area only containing eye is then taken as the second template. From the second template, an area only containing eye ball is taken as the third template. After obtaining the twelve templates, each template is used to mask each eye-like region by sliding the template pixel by pixel in the region to find the best matched area. For

Table 1. The templates for six eye classes.

Eye Type	Template Images					
Normal						
High light						
Low light						
Left-side						
Right-side						
Bias lighting						

each eye-like region, the area with the largest matching value among those of the best matched areas in a class is considered as a candidate eye area. Thus, each eye-like region will have three candidate eye areas. Note that covariance is used to measure the matching degree and is defined as follows:

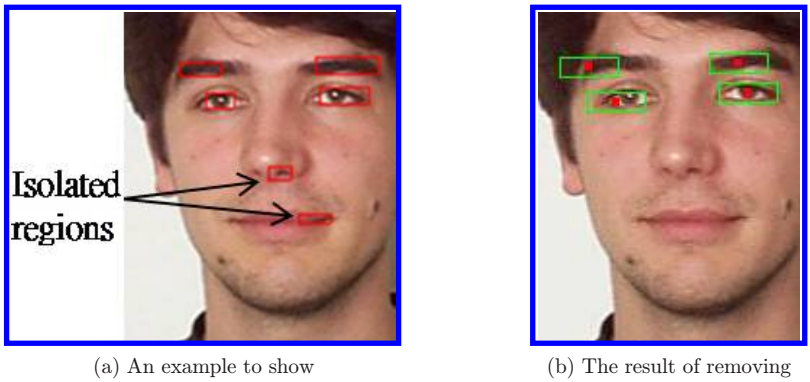
$$\mathrm{cov}(A, B) = \frac{\sum_{i=1}^{n}(a_i - A_{\mathrm{mean}})(b_i - B_{\mathrm{mean}})}{n}, \tag{2}$$

where $A = [a_1, a_2, \ldots, a_n]$ and $B = [b_1, b_2, \ldots, b_n]$ are two areas, and A_{mean} and B_{mean} are the mean values for each area. Now, for each eye-like region, the center of the candidate eye with the highest Cb/Cr value (the most possible nonskin area) is considered as the best eye center of the eye-like region [see the red points in Fig. 12(b)]. If an eye-like region with the best eye center does not lie in the true eye, we use the previous method to take three new templates from the image containing the region. The procedure is repeated until all eye-like regions are processed. After the procedure is finished, we find that nine new templates (see Table 1) belonging to three new classes are extracted. These three new classes are on the left (the eye sees the left side), and the right side (the eye sees the right side) and bias lighting. Since the images in HHI have different face sizes, the extracted templates will also have different sizes.

Based on the extracted 21 templates, the above matching procedure is applied to each eye-like region in HHI and Champion databases. Thus, we can obtain the best eye center in each eye-like region. Finally, all eye-like regions are refined to be a $w \times h$ region centering at the best eye center. In the paper, we set w to 30, h to 10 [see the green rectangles in Fig. 12(b)]. After an eye-like region is refined, we judge if it is an isolated region. If yes, the region will be removed. Figure 13(b) shows the result of removing those refined isolated false regions.

3.3.3. *Forehead-hair false regions removal*

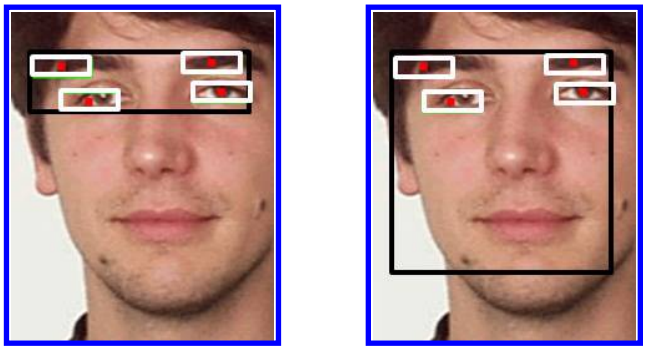
Before describing the proposed forehead-hair false region removing algorithm, we will define the bounding rectangle for a face. Let R be the minimum rectangle containing



(a) An example to show isolated regions.

(b) The result of removing the isolated regions in (a).

Fig. 13. The result of removing isolated false regions.



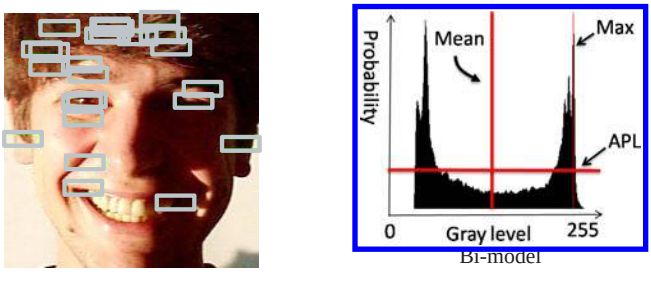
(a) The minimum rectangle containing all eye-like regions.

(b) The expanding result of (a).

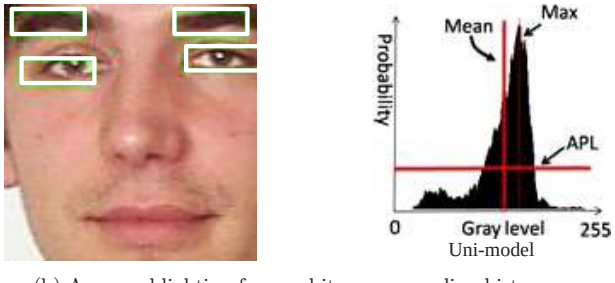
Fig. 14. An example to illustrate the bounding rectangle of a face.

all eye-like regions, then R is expanded in the vertical direction to form a square rectangle (see Fig. 14). The square rectangle is defined as the bounding rectangle of the face.

In general, pixels in hair or shadow regions caused by a bias lighting have lower gray levels than pixels in a face. Based on this property, we find that if the bounding rectangle of a face contains a part of hair or side shadow, its gray-level histogram, $h(x)$, will be a bimodel distribution; else the histogram will be a unimodel distribution (see Fig. 15). Note that a bimodel histogram must have a valley with value lower



(a) A bias lighting face and its corresponding histogram.



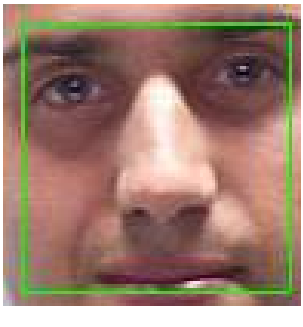
(b) A normal lighting face and its corresponding histogram.

Fig. 15. Two examples to show different models of gray level histograms.

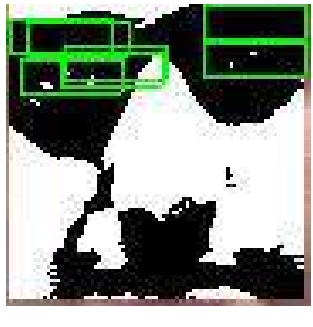
than the mean of the histogram. Based on this fact, to distinguish these two models, an average probability horizontal line (APL) $Y = T$ with $T = \frac{1}{256}\sum_{x=0}^{255} h(x)$ is first defined. If the number of the intersection line segments between the APL and the histogram is two, a bimodel distribution is identified [see Fig. 15(a)]. Otherwise, if only one line segment exists, a unimodel distribution is identified [see Fig. 15(b)]. For a face bounding rectangle, those eye-like regions appearing in the hair area should be removed. Based on the above discussion, a procedure is proposed to remove forehead-hair false regions. To identify the hair part, we set the initial bilevel threshold $t0$ as average gray value of the face rectangle for a bimodel histogram. If the face is a unimodel, we set the initial threshold $t0$ as $\min\{t|\mathrm{ACC}(t) > 5\% \times \sum_{x=0}^{255} h(x)\}$ with $\mathrm{ACC}(t) = \sum_{x=0}^{t} h(x)$. The pixels with gray levels less than $t0$ are labeled as black pixels. However, the threshold may be improper such that over-segmentation or under-segmentation (see Fig. 16) will occur. To treat this problem, the threshold $t0$ will be adjusted according to the following procedure. After applying the bilevel thresholding, if no two eye-like regions containing over O_{th} (30%) black pixels, an over-segmentation is detected [see Fig. 16(b)]. The $t0$ should be adjusted to be larger. The adjustment rule is to take the nearest right valley to the current $t0$ on the

(a) Original image

(b) An over-segmentation obtained after applying the threshold process to (a).



(c) Original image



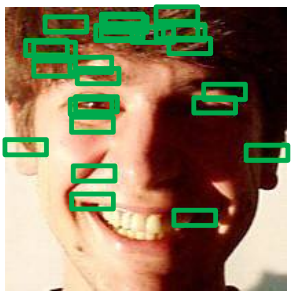
(d) An under-segmentation obtained after applying the threshold process to (c).

Fig. 16. Two examples to show the over-segmentation and under-segmentation.

histogram as a new threshold $t0$. If no valley is found, then $t0$ will be adjusted by increasing a threshold value (N_{th}), here N_{th} is set to be 5.

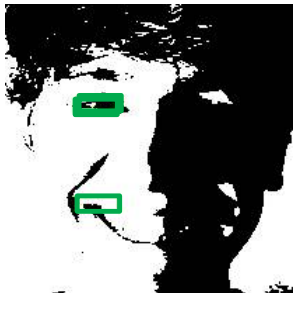
On the other hand, if all eye-like regions are located in a big black area, an under-segmentation is detected [see Fig. 16(d)], $t0$ will be refined to be smaller. The adjustment rule is to take the nearest left valley to the current $t0$ on the histogram as a new threshold $t0$. If no valley was found, then the $t0$ will be decreased by N_{th}. The above procedure is repeated until no over- or under-segmentation occurs. In order to prevent a dangling situation, any used valley will be removed to avoid reusage.

Then the bilevel thresholding operator with the final threshold is applied to identify the dark regions in the face bounding rectangle [see Fig. 17(b)]. If a dark area is hair, it should be large in the face bounding rectangle. If a dark area is iris, it should be small. So, for a dark region with area larger than $\frac{1}{4}$ face bounding rectangle, we consider it as a hair region, and all eye-like regions in the hair region are removed. In addition, if a dark region has the ratio of the height over the width larger than a threshold 2.5, we will remove the region. On the other hand, since the iris and eyelids have lower gray level than the other part of an eye, if an eye-like region contains less than 3% dark pixels, the region is also removed. Note that any eye-like region located at the shadow part will also be removed by the above procedure [see Fig. 17(c)]. To resume these regions, we use a previously mentioned fact if an eye is found, then the



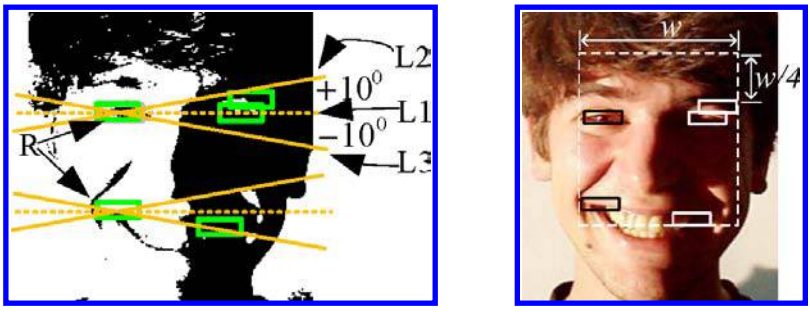
(a) The eye-like regions before applying the removing procedure.

(b) The result of applying the bilevel threshold to (a).



(c) The result of removing the forehead-hair false regions in (b).

Fig. 17. The result of applying the forehead-hair false region removing procedure.



(a) The bounded ranges for resuming the removed eye-like regions in the shadow area of a face.

(b) Three eye-like regions (marked by white rectangles) retrieved.

Fig. 18. An example to explain the eye-like region resuming procedure.

other eye should be found on its right side or left side. For each remaining eye-like region, R, we first draw three lines, *L1*, *L2* and *L3*, passing through the center of R. *L1* is a horizontal line, the angles between $L1$ and $L2$ (or $L3$) is a predefined angle A_{th} (here, $\pm 10^{\circ}$ is used) [see Fig. 18(a)] for a bimodel histogram face. Then a bounded range, BR, is defined to be the area bounded by $L2$ and $L3$. Based on the bounded range, all eye-like regions in the bounded range removed by the forehead-hair false region removing procedure will be retrieved [see Fig. 18(b)]. Now the remaining eye-like regions are used to form a new bounding rectangle [see the dotted rectangle in Fig. 18(b)] which will be used to determine the true horizontal eye line.

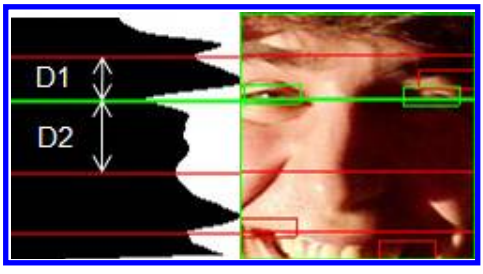
4. The Proposed Horizontal Eye Line Detector

In this section, we will propose a novel eye line detection algorithm. Because eyes and eyebrows are regions with low gray-level, their locations will correspond to the local valley of the horizontal projection histogram. So the eye line detection algorithm can be reduced as a valley finding procedure on the horizontal gray value projection histogram. In the valley finding procedure, all valleys are taken first, then the small valleys with value smaller than a threshold D_{th} (here, we set D_{th} to be 3) will be removed. If some valleys are located at similar locations with distance between two neighboring ones smaller than a threshold B_{th} (here, B_{th} is set to be 3), the deepest valley will be kept, others are discarded.

The details are described as follows. For a given bounding rectangle of a face, if there is only one valley in the projection histogram and there is a pair of eye-like regions near the valley [see Fig. 19(a)], then the horizontal line [the green line in Fig. 19(a)] is defined as the line passing through the vertical middle location between the two eye-like regions. If there are two valleys, and if the distance between these two valleys is less than a threshold and there are at least one pair of eye-like regions near the two horizontal lines passing through these two valleys, respectively, then the upper one [the red line in Fig. 19(b)] is considered as the eyebrow line and the other

(a) Only one valley case.

(b) Two valleys case.



(c) $D1 < s1 \times D2$ with two eye-like regions near the second valley line.

(d) $D1 > s2 \times D2$ with two eye-like regions near the first valley line.

Fig. 19. Some examples for the horizontal projection.

[the green line in Fig. 19(b)] is the eye line. If there are three or four valleys, we will execute the following procedure (called procedure A). First, we define the distance between the first and second valleys as $D1$ and between the second and third valleys as $D2$. Then we check $D1$ and $D2$ to see if they satisfy the following two conditions.

$$\text{Condition 1}: \quad D1 < s1 \times D2.$$
$$\text{Condition 2}: \quad D1 > s2 \times D2.$$

Note that $s1$ and $s2$ are set as 1.2 and 3.0 in the paper.

If valleys satisfy Condition 1 and there exist two eye-like regions near the second valley line, then the second valley line [the green line in Fig. 19(c)] is considered as the eye line. On the other hand, if Condition 2 is satisfied and there exist two horizontal eye-like regions near the first valley line, then the first valley line [the green line in Fig. 19(d)] is considered as the eye line. If both conditions are not satisfied, we will apply procedure A described above to the next three valleys (the second, third and fourth). If no eye line is found, we conclude that the region is not a face. If an eye line is found, any eye-like region with distance from the eye line larger then 1.5 height of the eye-region will be removed. Then, the remaining regions will be used to outline a face rectangle where the left (right) position is set to be the leftmost (rightmost) position of the leftmost (rightmost) region. The top position of the rectangle is defined as the position above three times height of an eye-like region of the upper location of the topmost region. And the height of the rectangle is defined as 1.3 times width of the rectangle. Figure 20 shows an example of the located face location.

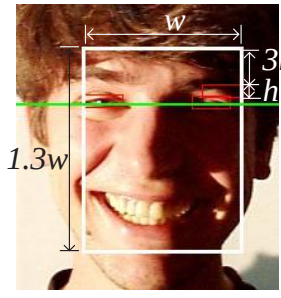


Fig. 20. The detected face location based on the eye-like regions.

5. Experimental Results

In order to show the effectiveness of the proposed method, we apply the proposed method to HHI face database[7] of 206 images (see Fig. 21) and Champion face database[16] of 227 images (see Fig. 22). We also collect some other images from Internet, MPEG7 video clips, PIE database[14] to evaluate the performance. These contain face images of different racial persons under different kinds of lighting conditions (such as overhead, side and color lightings). Note that one of our main applications of the method is for a drowsy driving warning system and in most cases, a driver's head position is usually frontal or near profile. Since the head is rarely in-plane rotated when people are driving, in this paper, we only detect the horizontal eye line and allow in-plane rotation with A_{th} ($\pm 10^{\circ}$) degrees. The system is implemented by Java language on a laptop with Pentium(R) M processor 1.4 GHz CPU. Figure 22 shows some successful results of applying our method to some faces

Fig. 21. Parts of images from HHI face database with different poses and eye-glasses.

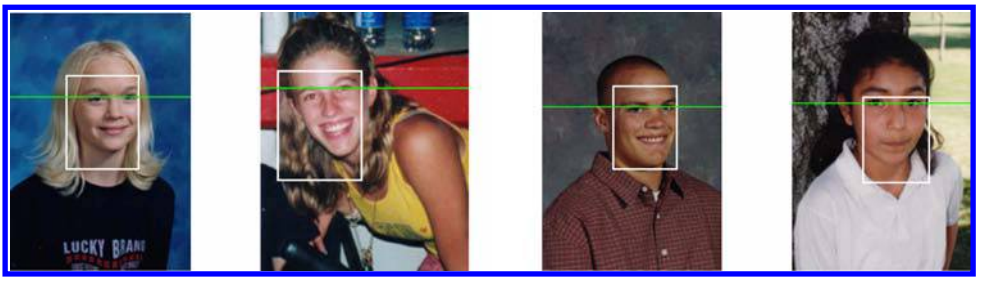


Fig. 22. Parts of images from Champion face database with different skin colors.

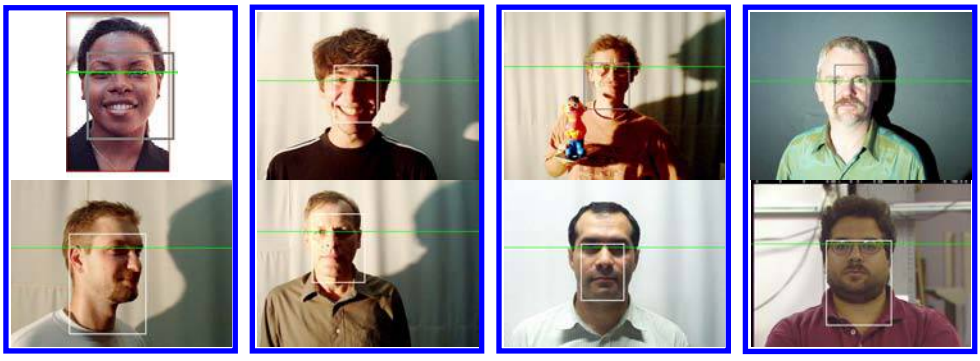


Fig. 23. The detection results for persons with shadows and various lighting sources on their faces.

with different skin colors. The successful results for faces with different poses and eye glasses are shown in Fig. 21. Even when there is shadow on a face, the eye line can be detected (see Fig. 23).

If the detected eye line is located under eye brow and near eyes, we consider it to be a correct detection; otherwise it is an error detection. The tolerance value is the height of an eye-like region. If no horizontal eye line is detected for a face image, we call it a missing detection. Figure 24 shows the successful detection results for a set of images from the Internet, MPEG7 video clips and PIE database images.

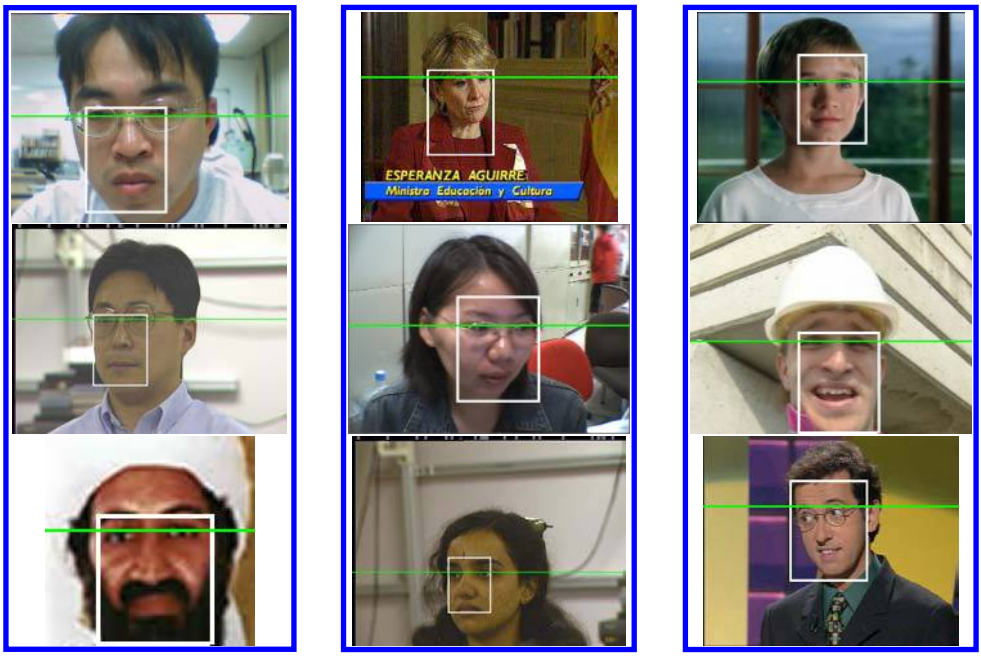


Fig. 24. The detection results of face images collected from the Internet, MPEG7 video clips and PIE database.

Table 2. Comparison results on HHI database (image size 640 × 480).

Head Pose		Frontal	Near Frontal	Half Profile	Profile	Total
No. of Images		66	54	75	11	206
Grouped Skin Regions						
No. of false positive	Proposed method	1345	936	1540	969	4790
	Hsu[4]	3145	2203	3781	277	9406
Average execution time per image	Proposed method	0.84 ± 0.62 sec (Intel(R) Pentium(R) M processor 1.4 GHz)				
	Hsu[4]	1.56 ± 0.45 sec (Intel(R) Pentium(R) processor 1.7 GHz)				
Rectangle Merge						
No. of false positive	Proposed method	31	58	93	5	187
	Hsu[4]	468	287	582	39	1376
Average execution time per image	Proposed method	< 0.01 sec				
	Hsu[4]	0.18 ± 0.23 sec				
Facial Feature Detection						
No. of false positive	Proposed method	1	0	2	2	5
	Hsu[4]	4	6	14	3	27
Correct detection rate	Proposed method	96.97%	98.15%	96%	72.73%	95.63%
	Hsu[4]	89.40%	90.74%	74.67%	18.18%	80.58%
Average execution time per image	Proposed method	7.065 sec				
	Hsu[4]	22.97 ± 17.35 sec				

Table 3. Comparison results on Champion database.

Head Pose	Frontal					
No. of Images	227					
	Grouped Skin Regions		Rectangle Merge		Eye Line Detection	
	Proposed Method	Hsu[4]	Proposed Method	Hsu[4]	Proposed Method	Hsu[4]
No. of false positive	490	5582	209	382	7	14
Correct detection rate	100%	99.12%	100%	99.12%	94.27%	91.63%
Average execution time per image	0.11 ± 0.09 sec	0.08 ± 0.036 sec	< 0.01 sec	0.08 ± 0.036 sec	3.984 sec	5.78 ± 4.98 sec

For HHI database, the correct detection rate is 95.63% and the correct detection rate on Champion database is 94.27%. Since the method in Ref. 4 also uses HHI and Champion databases to do experiments, here, we list the comparison results in Tables 2 and 3. From these tables, we can see that the speed of the proposed method is higher and the detection rate is better in both databases. Figure 25 shows the error and missing detection results.

Fig. 25. Some error/missing examples.

6. Conclusions

In this paper, we have developed a horizontal eye line detector for face images under various environments. In order to handle the environmental factors, the detector combines the intensity and color information to explore eye positions and to locate the eye line location. The experimental results showed that the proposed method is robust over a wide range of lighting conditions, different poses and races. Even when there is shadow on a face, the eye line can be detected. The proposed eye line detector has a highly accurate detection rate and can be further used to locate the true positions of eyes and face under various environments.

Acknowledgments

This research was supported in part by the National Science Council of R.O.C. under contract NSC-95-2221-E-009-364 and Taiwan Information Security Center at NCTU.

References

1. T. Y. Chow, K. M. Lam and K. W. Wong, Efficient color face detection algorithm under different lighting conditions, *J. Electron. Imag.* **15**(1) (2006) 013015(1)−013015(10).
2. T. Hayami, K. Matsunaga, K. Shidoji and Y. Matsuki, Detecting drowsiness while driving by measuring eye movement — A pilot study, *IEEE Proc. 5th Int. Conf. Intelligent Transportation Systems* (2002), pp. 156−161.
3. E. Hjelmas and B. K. Low, Face detection: A survey, *Comput. Vis. Imag. Underst.* **83**(3) (2001) 236−274.
4. R. L. Hsu, M. A. Mottaleb and A. K. Jain, Face detection in color images, *IEEE Trans. Patt. Anal. Mach. Intell.* **24**(5) (2002) 696−706.
5. P. Kakumanu, S. Makrogiannis and N. Bourbakis, A survey of skin-color modeling and detection methods, *Patt. Recogn.* **40**(3) (2007) 1106−1122.
6. J. Miao, B. Yin, K. Wang, L. Shen and X. Chen, A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template, *Patt. Recogn.* **32**(7) (1999) 1237−1248.
7. MPEG7 content set from Heinrich Hertz Institute, http://www.darmstadt.gmd.de/mobile/hm/projects/MPEG7/Documents/N2466.html, Oct. 1998.
8. V. Perlibakas, Automatical detection of face features and exact face contour, *Patt. Recogn. Lett.* **24**(16) (2003) 2977−2985.

9. H. Rowley, S. Baluja and T. Kanade, Neural network-based face detection, *IEEE Trans. Patt. Anal. Mach. Intell.* **20**(1) (1998) 23−38.
10. S. Satoh, Y. Nakamura and T. Kanade, Name-it: Naming and detecting faces in news videos, *IEEE Multimed.* **6**(1) (1999) 22−35.
11. D. Saxe and R. Foulds, Toward robust skin identification in video images, *Proc. Second Int. Conf. Automatic Face and Gesture Recognition* (1996), pp. 379−384.
12. F. Y. Shih, S. Cheng, C. F. Chuang and P. S. P. Wang, Extracting faces and facial features from color images, *Int. J. Patt. Recogn. Artif. Intell.* **22**(3) (2008) 515−534.
13. F. Y. Shih and C. F. Chuang, Automatic extraction of head and face boundaries and facial features, *Inform. Sci.* **158** (2004) 117−130.
14. T. Sim, S. Baker and M. Bsat, The CMU pose, illumination, and expression database, *IEEE Trans. Patt. Anal. Mach. Intell.* **25**(12) (2003) 1615−1618.
15. J. Song, Z. Chi and J. Liu, A robust eye detection method using combined binary edge and intensity information, *Patt. Recogn.* **39**(6) (2006) 1110−1125.
16. The Champion database, http://www.libfind.unl.edu/alumni/events/breakfastt_for_champions.htm, Mar. 2001.
17. M. Turk and A. Pentland, Eigenfaces for recognition, *J. Cogn. Eurosci.* **3**(1) (1991) 71−86.
18. H. Wang, P. Li and T. Zang, Boosted Gaussian classifier with integral histogram for face detection, *Int. J. Patt. Recogn. Artif. Intell.* **21**(7) (2007) 1127−1139.
19. C. A. Waring and X. Liu, Face detection using spectral histograms and SVMs, *IEEE Trans. Syst. Man Cybern. — Part B: Cybern.* **35**(3) (2005) 467−476.
20. J. Wu and Z. Zhou, Efficient face candidates selector for face detection, *Patt. Recogn.* **36** (5) (2003) 1175−1186.
21. M. H. Yang, D. J. Kriegman and N. Ahuja, Detecting faces in images: A survey, *IEEE Trans. Patt. Anal. Mach. Intell.* **24**(1) (2002) 34−58.

Min-Quan Jing received the M.S. degree in computer and information science in 1999 and Ph.D degree in computer science both from National Chiao Tung University in 2009.

His current research interests include image processing, pattern recognition and face detection.

Ling-Hwei Chen received the B.S. degree in mathematics and the M.S. degree in applied mathematics from National Tsing Hua University, Hsinchu, Taiwan in 1975 and 1977, respectively, and the Ph.D. degree in computer engineering from National Chiao Tung University, Hsinchu, Taiwan in 1987. From August 1977 to April 1979 she worked as a research assistant in the Chung-Shan Institute of Science and Technology, Taoyan, Taiwan. From May 1979 to February 1981, she worked as a research associate in the Electronic Research and Service Organization, Industry Technology Research Institute, Hsinchu, Taiwan. From March 1981 to August 1983, she worked as an engineer in the Institute of Information Industry, Taipei, Taiwan. She is now a director of the Institute of Mutimedia Engineering at the National Chiao Tung University.

Her current research interests include image processing, pattern recognition, mutimedia retrieval and multimedia steganography.