Proceedings of the 2004 IEEE
International Conference on Networking, Sensing & Control
Taipei, Taiwan, March 21-23, 2004

# Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching

**Wen-Bing Horng, Chih-Yuan Chen, Yi Chang**
Department of Computer Science
and Information Engineering
Tamkang University
Taipei, Taiwan, R.O.C.
horng@mail.tku.edu.tw

**Chun-Hai Fan**
Department of Transportation Management
Tamkang University
Taipei, Taiwan, R.O.C.
chunhai@mail.tku.edu.tw

**Abstract-** *A vision-based real-time driver fatigue detection system is proposed for driving safety. The driver's face is located, from color images captured in a car, by using the characteristic of skin colors. Then, edge detection is used to locate the regions of eyes. In addition to being used as the dynamic templates for eye tracking in the next frame, the obtained eyes' images are also used for fatigue detection in order to generate some warning alarms for driving safety. The system is tested on a Pentium III 550 CPU with 128 MB RAM. The experiment results seem quite encouraging and promising. The system can reach 20 frames per second for eye tracking, and the average correct rate for eye location and tracking can achieve 99.1% on four test videos. The correct rate for fatigue detection is 100%, but the average precision rate is 88.9% on the test videos.*

**Keywords:** driver fatigue detection, eye detection, face detection, template matching.

## 1 Introduction

It is a hard test of endurance for drivers to take long distance driving. It is very difficult for them to pay attention to driving on the entire trip unless they have very strong willpower, patience, and persistence. Thus, the *driver fatigue* problem has become an important factor of causing traffic accidents. A driver fatigue detection system can detect whether the driver is tired, such as dozing or inattention, so as to generate some warning alarms to alert the driver. Therefore, the system can reduce not only traffic accidents but also the social cost caused by these accidents. Because human eyes express the most direct reaction when dozing, eye blinking is usually used as the basis for driver fatigue detection by researchers.

In recent years image processing on human faces has been used in many applications, such as face recognition, face analysis [4], eye detection [6], gaze tracking [9], etc. Among all of these researches, the first step is usually to locate faces. Recently, the human face detection techniques have matured gradually. The detection techniques can be divided into two major categories. The first is based on face features [4][6]. The second is based on face colors [1][7-11]. Feature-based face detection methods utilize some well-known knowledge on human faces, such as the shape of the face, the relative locations of the eyes, the nose, and the mouth in a face, and so forth, to judge whether the observed features on an image satisfy such criteria. On the other hand, color-based face detection methods build on specific color models to locate faces based on skin colors. Since skin colors have quite stable distribution in some color models, face detection based on skin colors becomes more popular. In this paper, the HSI color model is used to locate faces since face colors have fixed distribution range on the hue component of the HSI model, decreasing the influence of brightness changes.

When observing a facial image, the most perceptible facial features are usually the places with obvious edges, particularly the ocular outlines. Since eyes have the nature of sophisticated edges, many researches use this attribute for eye detection [1][2][4][7]. In this paper the Sobel edge operator and projection technique are used for eye location. The obtained eyes' images are used as the dynamic templates for eye tracking in the next frame. Besides, they are also used to determine whether the eyes are open or closed, which is the basis of driver fatigue detection in this research.

The paper is organized as follows. Related researches on driver fatigue detection are briefly surveyed in section two. The proposed vision-based automatic driver fatigue detection system, including face detection, eye detection, eye tracking, and fatigue detection, is presented in section three. The experiment results of the system and conclusion are given in the last section.

## 2 Related works

In the past ten years, many countries all over the world have begun to pay attention to the driving safety problem and to investigate the drivers' mental states relating to driving safety. Also, warning systems for drivers are also presented [1][2][7]. In 1997 and 1999, Eriksson and Papanikolopoulos [2] and Singh and Papanikolopoulos [7], respectively, proposed two papers on driver fatigue detection based on image processing techniques for driving safety.

In [2], the authors used the symmetric property of faces to detect facial area on an image. Then, they used pixel difference to find the edges on the facial region to locate the vertical position of the eyes. Since the edge detection they used cannot clearly mark edges, it is not easy to locate accurate ocular locations. The authors used thresholding to improve the location of the ocular places. After finding the approximate eyes positions, a concentric circle template was designed to locate the exact eyes locations, and the template was used to track eyes in the following images. Face symmetry is an obvious feature for an upright face. However, it usually fails to locate the correct face position when the face tilts, rotates, or is shadowed.

Instead of using symmetric central line method, the authors in [7] used the Gaussian distribution of the skin colors to distinguish skin and non-skin pixels [10][11] for face detection. The Sobel vertical edge operator was used for eye detection. Besides, they built a database of eye images as the templates for eye detection and tracking. Although the Gaussian distribution of skin colors based on the RGB color model was used to predict skin quite well, the method cannot get rid of the factor of brightness changes. In addition, the eye images in the database as templates may be quite different from drivers' eyes, which will reduce the accuracy for eye location.

In order to cope with the problem of intensity changes, the HSI color model is adopted to represent face skin in this paper since the model detaches the intensity from the hue of a color. Furthermore, the obtained eyes' images are used as the dynamic templates for eye tracking in the paper, instead of using the static eye images in the database as used in the above method, to enhance the reality and reliability of the templates so as to increase accuracy of eye tracking.

# 3 Driver fatigue detection system

The system uses a color camera mounted on the dashboard of a car to capture the images of the driver for driver fatigue detection. The flow chart of the proposed fatigue detection system is depicted in Figure 1. The first image is used for face location and eye detection. If any one of these detection procedures fails, then go to the next frame and restart the above detection processes. Otherwise, the subsequent images are used for eye tracking based on the obtained eye images in the current image as the dynamic templates. If eye tracking fails, the processes of face location and eye detection restart on the present image. These procedures continue until there are no more frames. The detailed steps are described in the following subsections.

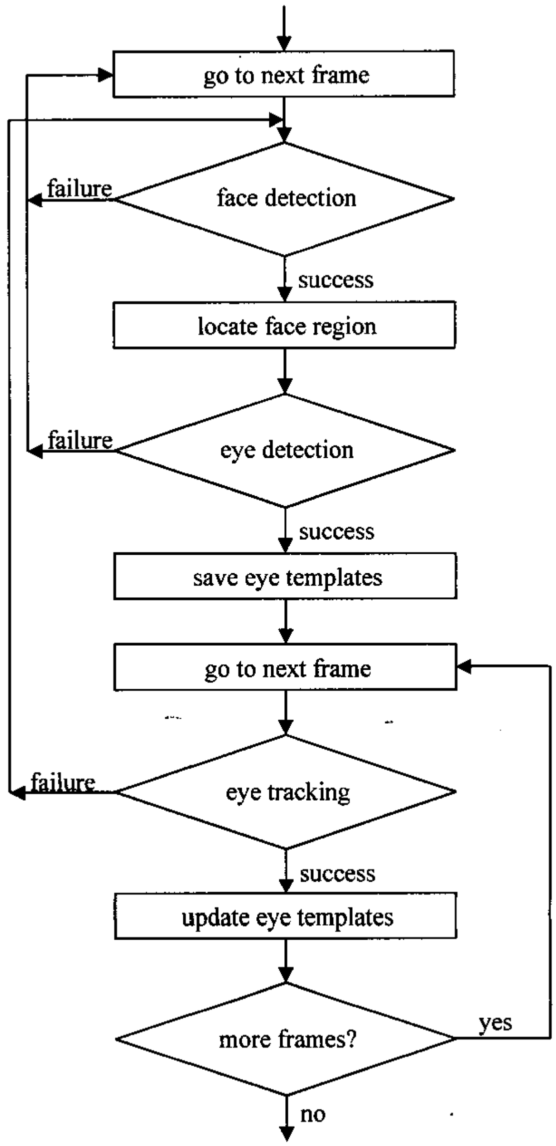



Figure 1. Flow chart of the driver fatigue detection system

## 3.1 Face detection

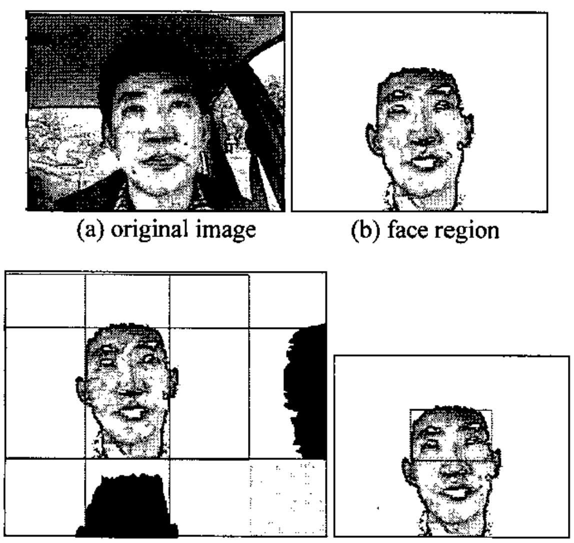
Digital images usually adopt the RGB color space to represent colors. However, any color in the RGB space not only displays its hue but also contains its brightness. For two colors with the same hue but different intensities, they would be viewed as two different colors by the human visual system. In order to accurately distinguish skin and non-skin pixels so that they will not be affected by shadows or light changes, the brightness factor must be excluded from colors. Since in the HSI color model hue is independent of brightness, this model is well suited for distinguishing skin and non-skin colors no matter whether the face is shadowed or not. Thus, it is used in this paper for face detection. The HSI model has three components: hue, saturation, and intensity (or brightness). The detailed method of converting a color between the RGB space and the HSI model refers to [3]. The following equations (1) to (3) show the results of converting a color from the RGB model to the HSI model. For any color with three elements, $R$, $G$, and $B$, normalized in the range [0, 1], the formulas for converting to its corresponding $H$, $S$, and $I$ components are given as follows.

$$H = \cos^{-1}\left\{ \frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}} \right\} \quad (1)$$

$$S = 1 - \frac{3}{R+G+B}\min(R,G,B) \quad (2)$$

$$I = \frac{R+G+B}{3} \quad (3)$$

When $(B/I) > (G/I)$, $H = 360^{\circ} - H$. From the face images of the test videos, it is observed that the $H$ values of skin colors fall in the range of $[10^{\circ}, 45^{\circ}]$. By using this condition, it is easy to separate the face region, as shown in Figure 2(b). By performing the vertical projection on skin pixels, the right and left boundaries can be found when the projecting values are exceeding a threshold. Similarly, by performing the horizontal projection on skin pixels within the left and right boundaries, the top and bottom boundaries can also be found, as shown in Figure 2(c). According to the normal position of the eyes in a face, it is reasonable to assume that the possible location of eyes will be in the block of the upper two fifths of the face region, as shown in Figure 2(d).



(a) original image (b) face region

(c) projection (d) eye region

Figure 2. Result of face detection

## 3.2 Eye detection

After defining the eye region, the original color eye region is converted into gray scale by formula (4).

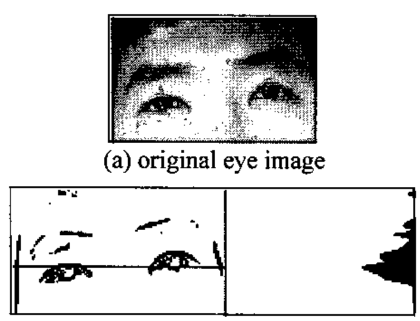
$$Y = 0.299R + 0.587G + 0.114B \quad (4)$$

where $Y$ is the gray value of a color ($R$, $G$, $B$). Then, the Sobel edge operator is used to compute the edge magnitude in the eye region to find the vertical position of the eyes. In order to save computing time, an approximate calculation of the edge magnitude, $G(x, y)$, is computed in equation (5) as follows.

$$G(x,y) \approx |S_x| + |S_y|, \quad (5)$$

where $S_x$ and $S_y$ are the horizontal and vertical gradient values obtained, respectively, from the Sobel horizontal and vertical edge operators. For detailed calculation, refer to [3][5].

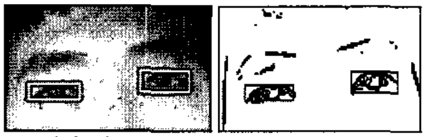
After computing the edge magnitude of each pixel in the eye region, an edge map is obtained by defining a pixel $(x, y)$ being an edge pixel if its edge magnitude $G(x, y)$ is greater than some threshold, as shown in the left part of Figure 3(b). Performing horizontal projection on the edge map, the exact vertical position of the eyes can be located at the peak, as shown in the right part of Figure 3(b). Sometimes, it may need to smooth horizontal projections for locating peaks. After locating the vertical position of the eyes, the left and right eye positions can be located by finding the largest connected components from the center of the eye region along the horizontal line of the vertical position of the eyes. In order to avoid marking noise, the total number of pixels in a connected component must exceed some threshold so that it can be regarded as a reasonable eye. After finding the connected component of an eye, a bounding box for the connected component on the original image is used to enclose the eye image, which is used as the template for eye tracking in the next frame, as shown in Figure 4.



(a) original eye image

(b) edge map and horizontal projection

Figure 3. Result of edge detection



(a) original eye image (b) edge map

Figure 4. Result of eye detection

### 3.3 Eye tracking

After finding the eye templates, they are used for driver's eye tracking by template matching. The searching area for matching is the original bounding box of an eye by expanding 10 pixels in each of four directions. Consider an eye template $g(x, y)$ locating in the position $(a, b)$ of the face image $f(x, y)$. Equation (6) is used for template matching.

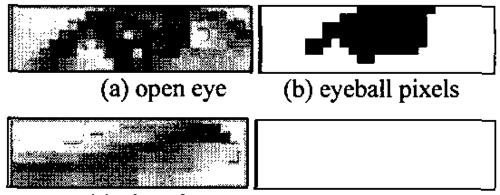
$$M(p,q) = \min_{p,q}\left[\sum_{x=0}^{x<w}\sum_{y=0}^{y<h}\left|f(x+p, y+q) - g(x,y)\right|\right] \quad (6)$$

where $w$ and $h$ are the width and height of the eye template $g(x, y)$, and $p$ and $q$ are offsets of the $x$-axis and $y$-axis in which $a - 10 \leqq p \leqq a + 10$ and $b - 10 \leqq q \leqq b + 10$. If $M(p^*, q^*)$ is the minimum value within the search area, the point $(p^*, q^*)$ is defined as the most matching position of the eye. Thus, $(a, b)$ is update to be the new position $(p^*, q^*)$. The new eye template is updated accordingly for eye tracking in the next frame.

If the distance between two newly updated eye templates does not satisfy certain constrain of the eyes, the eye tracking is regarded as failure. Then, the system will restart the face and eye location procedures.

### 3.4 Fatigue detection

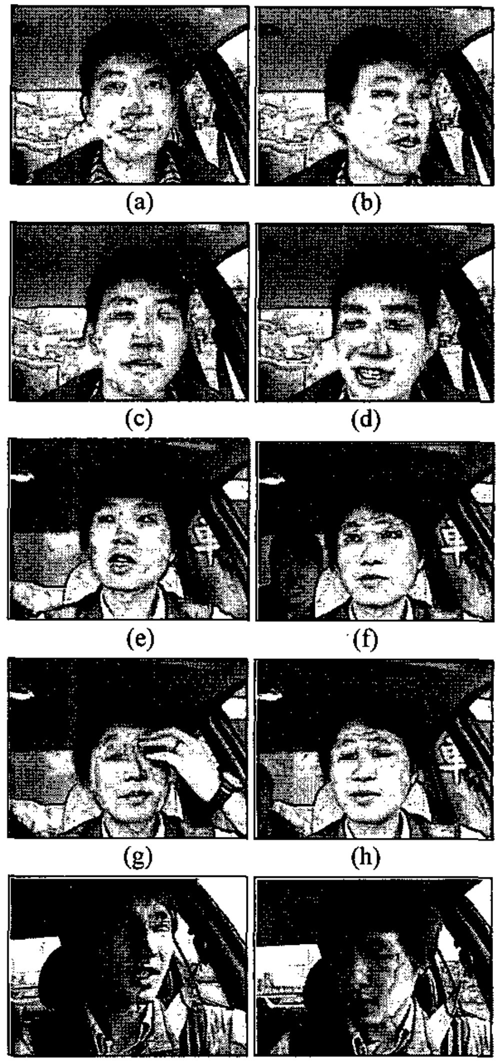
At this stage, the colors of the eyeballs in the eye templates are used directly for fatigue detection. Since the property that the eyeball colors are much darker is a quite stable feature, the eye templates are inverted (negated) and then converted to the HSI color model. The original darker eyeballs become brighter ones in the inverted image. According to the observation, the saturation values of eyeball pixels normally fall between 0.00 and 0.14. This observation is used to distinguish whether a pixel in an eye template is viewed as an eyeball pixel, as shown in Figure 5. When the eyes are open, there are some eyeball pixels, as shown in Figures 5(a) and 5(b). When the eyes are closed, there are no eyeball pixels, as shown in Figures 5(c) and 4(d). By checking the eyeball pixels, it is easy to detect whether the eyes are open or closed. In this system, it is defined that when the eyes close over 5 consecutive frames, then the driver is regarded as dozing.



(a) open eye (b) eyeball pixels

(c) closed eye (d) no eyeball pixels

Figure 5. Eyeball detection



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 6. Snapshots from the system during tracking under different people and different background

## 4 Experiment results

The proposed driver fatigue detection system uses a Sony PC115 color DV camera to capture driver's images. The system is tested under the environment of Pentium III 550 CPU and 128 MB RAM. The format of input video is 320x240 true color. After starting the system, it takes 67 milliseconds for face location and eye detection. Once the eye templates are found, eye tracking can achieve 20 frames per second. Recall that in this system, if the driver closes his/her eyes over 5 consecutive frames, then the driver is regarded as dozing.

Table 1 lists the results of eye tracking from four test videos, as shown in Figure 6. The field *Total Frame* means the total number of frames in each video. *Tracking Failure* is the count of eye tracking failure. And *Correct Rate* of eye tracking is defined as in equation (7), which is ratio of (*Total Frame* – *Tracking Failure*) to *Total Frames*.

$$Correct\ Rate = \frac{Total\ Frames - Tracking\ Failure}{Total\ Frames} \quad (7)$$

As can be seen from Table 1, the correct rate of eye tracking is higher than 98.3%, and the average correct rate can achieve 99.1%.

Table 1. Result of eye tracking

| | Video 1 | Video 2 | Video 3 | Video 4 |
|---|---|---|---|---|
| *Total Frames* | 2634 | 1524 | 2717 | 433 |
| *Tracking Failure* | 8 | 6 | 46 | 7 |
| *Correct Rate* | 99.7% | 99.6% | 98.3% | 98.4% |
| Average Correct Rate | | 99.1% | | |

Table 2. Result of fatigue detection

| | Video 1 | Video 2 | Video 3 | Video 4 |
|---|---|---|---|---|
| *Close Eyes* | 22 | 18 | 43 | 6 |
| *Real Dozing* | 3 | 4 | 15 | 2 |
| *Generate Warning* | 3 | 4 | 18 | 2 |
| *False Positive* | 0 | 0 | 3 | 0 |
| *False Negative* | 0 | 0 | 0 | 0 |
| *Correct Warning* | 3 | 4 | 15 | 2 |
| *Correct Rate* | 100% | 100% | 100% | 100% |
| *Precision Rate* | 100% | 100% | 83.3% | 100% |
| Average Precision Rate | | 88.9% | | |

Table 2 shows the result of driver fatigue detection on the four test videos. The field *Close Eyes* represents the number eye closing and then opening of the driver in the video. *Real Dozing* is the number of dozing determined by humans. *Generate Warning* is the number of fatigue detection which generates a warning alarm. *False Positive* is the number of wrongly generated warning alarms. *False Negative* is the number of not detected fatigue. *Correct Warning* is the number of correctly detected fatigue, which is equal to *Generate Warning* minus *False Positive* and *False Negative*. *Correct Rate* of fatigue detection is defined as in equation (8).

$$Correct\ Rate = \frac{Correct\ Warning}{Real\ Dozing} \quad (8)$$

And *Precision Rate* for fatigue detection is defined as in equation (9).

$$Precision\ Rate = \frac{Correct\ Warning}{Generate\ Warning} \quad (9)$$

As can be seen from Table 2, the system can correctly detect all fatigue states in all the test videos. In video 3, three mistakes made by the system are due to the interference of the driver's hand, as shown in Figure 6(g). In spite of this kind of interference, the average precision rate for fatigue detection for all videos can still achieve 88.9%.

## 5 Conclusion

This paper presents a vision-based real-time driver fatigue detection system for driving safety. The system uses the HSI color model to detect faces of input images and the Sobel edge operator to locate the eyes positions and to obtain eye images as the dynamic templates for eye tracking. Finally, the obtained eye images are converted to the HSI model to distinguish eyeball pixels to determine whether the eyes are open or closed for judging driver fatigue. The system is tested on a PC of Pentium III 550 CPU with 128 MB RAM. The average correct rate on four test videos for eye detection and tracking can reach 99.1%. On the four videos, the system can correctly detect all fatigues marked by humans. However, the precision rate for driver fatigue detection can also reach 88.9%. The speed of this system can reach up to 20 frames of size 320x240 per second for eye tracking.

## References

[1] M. Betke and W.J. Mullally, "Preliminary Investigation of Real-Time Monitoring of a Driver in City Traffic," *Proceedings of the IEEE International Conference on Intelligent Vehicles*, Dearborn, MI, pp. 563–568, October 2000.

[2] M. Eriksson and N. Papanikolopoulos, "Eye-Tracking for Detection of Driver Fatigue,"

*Proceedings of the International Conference on Intelligent Transportation Systems*, Boston, MA, pp. 314–319, November 1997.

[3] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Second Edition, Prentice-Hall, 2002.

[4] W.B. Horng, C.P. Lee, and C.W. Chen, "Classification of Age Groups Based on Facial Features," *Tamkang Journal of Science and Engineering*, Vol. 4, No. 3, pp. 183–191, 2001.

[5] R. Jain, R. Kasturi, and B.G. Schunck, *Machine Vision*, McGraw-Hill, 1995.

[6] R. Mariani, "Subpixellic Eyes Detection," *Proceedings of International Conference on Image Analysis and Processing*, Venice, Italy, pp. 496–501, September 1999.

[7] S. Singh, and N. Papanikolopoulos, "Monitoring Driver Fatigue Using Facial Analysis Techniques," *Proceedings of the International Conference on Intelligent Transportation Systems*, Tokyo, Japan, pp. 314–318, October 1999.

[8] S. Spors and R. Rabenstein, "A Real-Time Face Tracker for Color Video," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, Salt Lake, UT, pp. 1493–1496, May 2001.

[9] R. Stiefelhagen, J. Yang, and A. Waibel, "A Model-Based Gaze Tracking System," *Proceedings of the International Joint Symposia on Intelligence and Systems*, Rockville, MD, pp. 304–310, November 1996.

[10] J. Yang, W. Lu, and A. Waibel, "Skin-Color Modeling and Adaptation," *Proceedings of the Third Asian Conference on Computer Vision*, Vol. 2, Hong Kong, pp. 687–694, January 1998.

[11] J. Yang and A. Waibel, "A Real-Time Face Tracker," *Proceedings of the third Workshop on Applications of Computer Vision*, Sarasota, FL, pp. 142–147, December 1996.