Improved Driver Fatigue Detection System Based on Eye Tracking and Dynamic Template Matching

Wen-Bing Horng and Chih-Yuan Chen
Department of Computer Science and Information Engineering
Tamkang University
Taipei, Taiwan 25137, R.O.C.
E-mail: {horng@mail.tku.edu.tw}

Abstract*-In this article, we propose an improvement over our previous work on real-time driver fatigue detection system based on eye tracking and dynamic template matching. We utilize the 2D-log search and the three-step search algorithms, fast search algorithms used in the MPEG encoding technique, to improve the search efficiency of the original exhaustive search. The experimental results show that the 2D-log search and the three-step search take only 5.38% and 7.48% of the number of search points of the original exhaustive search. In addition, their correct rates of eye tracking are also slightly improved to be 96.81% and 97.23%, respectively, as compared to 96.01% of the original exhaustive search.*

Keywords: driver fatigue detection, eye tracking, fast search algorithm, intelligent transportation system, template matching.

1. Introduction

Driver fatigue has been one of the major causes of traffic accidents all over the world. In the UK, it is estimated that up to 20% of serious road accidents have resulted from driver fatigue, while in the US, there are around 30% of fatigue-related fatal accidents [17]. Therefore, many countries have invested lots of funds in building intelligent transportation systems to provide secure transportation, and researchers have begun to pay more attentions to the driving safety problem to decrease road crashes.

For improving driving safety, it can be roughly categorized into three approaches. One is to study the drivers' mental states relating to driving safety by psychologists [14, 15]. Another is to devise auxiliary equipments to improve driving safety by designing special car seats [2], by monitoring grip force change on the steering wheel [1], or by analyzing EEG (Electroencephalogram) recordings from sensors attached to the human body [18, 21]. The other is based on image processing techniques [5] to detect driver's fatigue to enhance driving safety. Some of these researchers utilized expensive infrared CCD cameras for easily locating eyes [6, 7, 10, 12, 22], while others employed ordinary CCD cameras for practical usage [3, 4, 16, 19, 20]. However, most of these image-based driver fatigue detection algorithms suffer from the illumination change problem. Besides, they might not be suitable for real-time applications due to their complicated computations in nature.

Recently, we have also proposed a vision-based real-time driver fatigue detection system based on ordinary CCD cameras to cope with the above deficiencies [9]. The system can be divided into four parts: face detection, eye detection, eye tracking, and fatigue detection. It was tested that the average correct rate of eye tracking could reach 96.01%, and the overall correct rate of driver fatigue detection of the system could achieve 100%.

However, during the eye tracking phase of our previous system, the exhaustive search with the conventional *mean absolute difference* (MAD) matching function is used. In this paper, we improve the efficiency of the original eye tracking method by using 2D-log search [11] and three-step search [13], the fast search algorithms used in video coding [8], to reduce search computation. The experimental results show that in the 2D-log and three-step searches for eye tracking, the number of search points can be greatly reduced than in the original exhaustive search. They are 5.38% and 7.48%, respectively, of the original search. Besides, their correction rates of eye tracking are also slightly improved.

The rest of the paper is organized as follows. In Section 2, we briefly review our previous driver fatigue detection system. In Section 3, we apply fast search algorithms to improve search efficiency for eye tracking. The experimental results are analyzed in Section 4. Finally, we conclude the paper in the last section.

2. Review of Our Previous System

In this section, we briefly review our previous driver fatigue detection system. For detailed information, please refer to [9]. The system consists of four components: face detection, eye detection, eye tracking, and fatigue detection. Figure 1 shows the flow chart of the driver fatigue detection system.

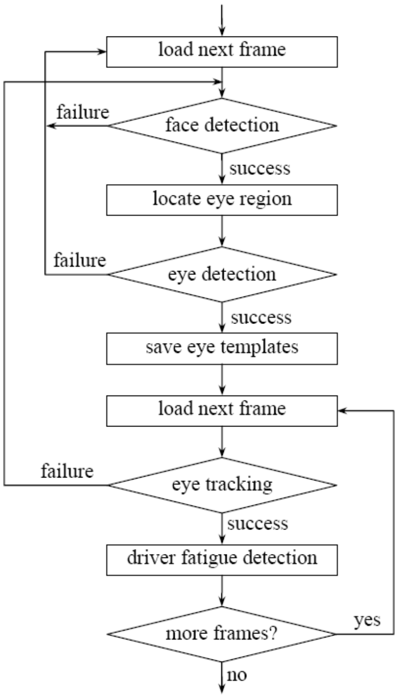


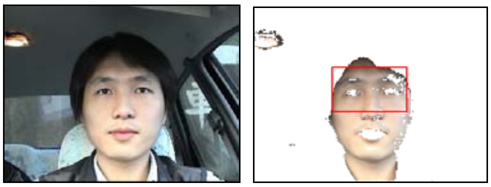
Figure 1. Flow chart of the driver fatigue detection system

At first, an ordinary color CCD camera is mounted on the dashboard of a car to capture the images of the driver for fatigue detection. The first frame is used for initial face detection and eye location. If any one of these detection procedures fails, then go to the next frame and restart the above detection processes. Otherwise, the current eye images are used as the dynamic templates for eye tracking on subsequent frames, and then the fatigue detection process is performed. If eye tracking fails, the face detection and eye location restart on the current frame. These procedures continue until there are no more frames.

2.1. 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, in this paper it is used for face detection.

The RGB color space used for representing color frames is first converted into the HSI color space for face detection to exclude the brightness factor from affecting skin color detection. A suitable range of hue values as well as horizontal and vertical projections can correctly detect the face region. The upper two fifths of the detected face region, called the *eye region*, is used for eye location. Figure 2(a) is a driver image. After performing face detection, the eye region enclosed in a bounding box is shown in Figure 2(b).



(a) Original image (b) Eye region

Figure 2. Result of face detection

2.2. Eye Detection

The original color information of the detected eye region is first converted into gray scale. Then, the Sobel edge operator is used for edge detection in the gray-level eye region, g_r, as follows. To reduce computation, an approximate edge magnitude, $mag(x, y)$, of a pixel (x, y) in g_r is computed as follows:

$$mag(x,y) = |S_1(x,y)| + |S_2(x,y)|$$

where $S_1(x, y)$ and $S_2(x, y)$ are the Sobel horizontal and vertical gradient values of pixel (x, y), respectively, and $|z|$ represents the absolute value of z.

The *edge map*, e_r, of the gray-level eye region g_r is defined by

$$e_r(x,y) = \begin{cases} black, & \text{if } mag(x,y) \geq T \\ white, & \text{otherwise} \end{cases}$$

where *black* and *white* stand for the black and white pixel values, respectively, and T for some threshold. Next, perform horizontal projection on the edge map e_r to find the vertical position of the eyes. Then, the left and right eye positions can be located by finding the largest connected components in e_r from the center. Finally, the eye subimages in the gray-level eye region g_r are located, which are used as the *dynamic eye templates* for eye tracking. Figures 3(a) and 3(b) show the gray-level eye region image and its corresponding edge map. After performing eye detection, two eye templates enclosed in bounding boxes are also shown in Figure 3.

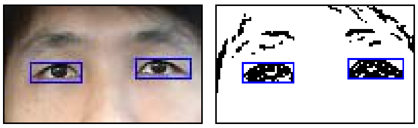

(a) Eye region (b) Edge map

Figure 3. Result of eye detection

2.3. Eye Tracking

Consider an eye template g_t of width w and height h, located at the position (a, b) in the original frame. The *search area* of a new frame for eye tracking is the eye template position by expanding some reasonable number of pixels in each of four directions: left, right, up, and down, as illustrated in Figure 4. Let d_{xmax} and d_{ymax} be the maximum displacements of the x-axis and y-axis, respectively. Thus, the size of the search area is $(w + 2d_{xmax})\times(h + 2d_{ymax})$, and the number of *search points* for the exhaustive search is equal to $(2d_{xmax} + 1)\times(2d_{ymax} + 1)$. This search area in the new color frame is first converted into a gray-level one, g_s, for eye tracking.

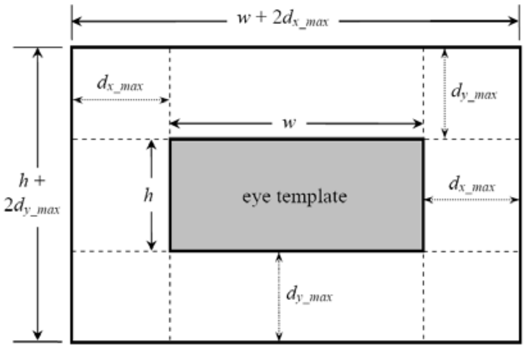


Figure 4. Search area of eye template

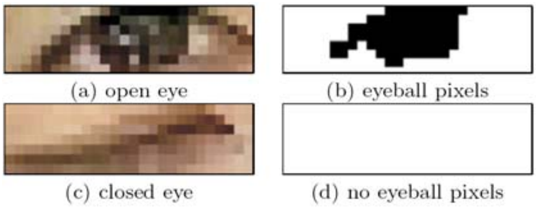
The following *mean absolute difference* (MAD) matching function is used for eye template matching:

$$M(p,q)=\sum_{x=0}^{w-1}\sum_{y=0}^{h-1}\left|g_t(x,y)-g_s(x+p,y+q)\right|$$

where p and q are displacements of the x-axis and y-axis, respectively, in which $(a - d_{xmax}) \le p \le (a + d_{xmax})$ and $(b - d_{ymax}) \le q \le (b + d_{ymax})$. If $M(p^*, q^*)$ is the minimum value within the search area, the point (p^*, q^*) is the most matching position of g_t, and let f_e denote the matched eye image in the current color frame for fatigue detection. Then, update the position (a, b) of g_t to be the new position (p^*, q^*) for tracking on subsequent frames.

2.4. Fatigue Detection

The stable feature of darker eyeball colors is used for fatigue detection. The matched f_e color image is first inverted (negated) and then converted into the HSI color space. Since the original darker eyeballs become brighter ones in the inverted image, pixels with low saturation values are regarded as eyeball pixels. Figure 5 shows the results of eyeball detection for an open eye image and a closed eye image. If the driver's eyes close over some consecutive frames, then he/she is regarded as dozing off, and a warming alarm is triggered to alert the driver.


(a) open eye (b) eyeball pixels

(c) closed eye (d) no eyeball pixels

Figure 5. Results of eyeball detection

Table 1 shows the experimental results of the driver fatigue detection system on five test videos. In these experiments, the driver is regarded as dozing off when his/her eyes close over 5 consecutive frames. In this table, the field n_1 stands for the number of frames, n_2 for the number of closed eyes, n_3 for the number of real dozing, n_4 for the number of detected dozing, n_5 for the number of correct dozing, n_6 for the correct rate of fatigue detection, and n_7 for the precision rate of fatigue detection, where $n_6 = n_5/n_3$ and $n_7 = n_5/n_4$. It is shown that the system could reach 100.0% correct rate of fatigue detection, while the precision rate could still achieve 89.3%.

Table 1. Results of fatigue detection

Video	n_1	n_2	n_3	n_4	n_5	n_6	n_7
1	2634	22	3	3	3	100.0%	100.0%
2	1524	18	4	4	4	100.0%	100.0%
3	2717	43	15	18	15	100.0%	83.3%
4	433	6	2	2	2	100.0%	100.0%
5	1399	3	1	1	1	100.0%	100.0%
Total	8707	92	25	28	25	100.0%	89.3%

3. Improved Eye Tracking

In our previous proposal, the exhaustive search with the MAD matching function is used for eye tracking. In this section, we apply the 2D-log search [11] and the three-step search [13], some fast search algorithms used in the MPEG encoding, to improve search performance.

As shown in Section 2.3, the exhaustive search needs to examine every search point within the search area, trying to find the best possible match. However, it requires a large amount of computations. In order to reduce the computational cost, several fast algorithms have been proposed at the price of slightly impaired performance. In general, a fast search algorithm starts with a rough search of a set of scattered search points. The distance between two nearby search points is called *step size*. At the end of each search step, the most promising search point becomes the new center point and another search step continues with probably a smaller step size. The above procedure is repeated until step size is equal to one, and the (local) optimum position is reached.

Note that as pointed out in [8, 11], if the matching function is monotonic along any direction away from the optimal point, it is guaranteed that a well-designed fast search algorithm can converge to the global optimal point. However, the real image signal is not a simple Markov process, and it contains coding and measurement noises. Therefore, the monotonic matching function assumption is often invalid, and consequently fast search algorithms are often suboptimal.

3.1. 2D-Log Search

The 2D-log search scheme was proposed by Jain and Jain [11]. It starts from the center point (zero displacement) of the search area to find the best match of the eye template based on some matching function. Figure 6 illustrates an example of the 2D-log search procedure, where circles represent the search points, and the number enclosed in each circle stands for the search step. In each search step, five search points in the search area with a diamond-shape are searched; they are the center point (the best matching point in the last step) and the four corner points of the diamond-shape with a fixed step size, r, as shown in Figure 6. Let d_{max} be the maximum search range in both x-axis and y-axis, i.e., $d_{max} = \max(d_{xmax}, d_{ymax})$. The initial value of r is set to $\max(2, 2^{m-1})$, where m = $\lfloor \log_2 d_{max} \rfloor$. The step size is reduced to one half when the best matching point is the center point or is the boundary point in the search area. Otherwise, the step size remains the same. The search ends with step size of one pixel, and nine search points (the center point and its eight neighbor points) are examined at this last step. Then, the best matching point at this final step is regarded as the best position of the eye.

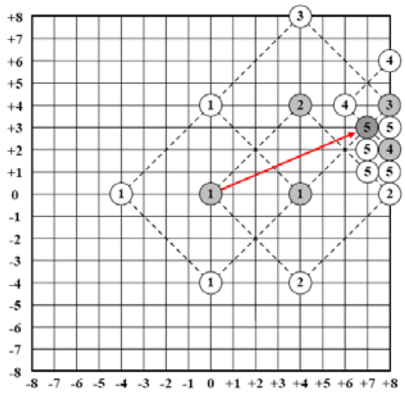


Figure 6. Illustration of 2D-log search

In the example of Figure 6 with $d_{max} = d_{x\_max} = d_{y\_max} = 8$, the 2D-log search requires 5 steps and 18 search points to reach the final destination with displacement (7, 3), where the best matching point in each search step is represented with a darker circle. In this example, the total computational cost for 2D-log search is much smaller than the exhaustive search, which requires $17^2 = 289$ search points.

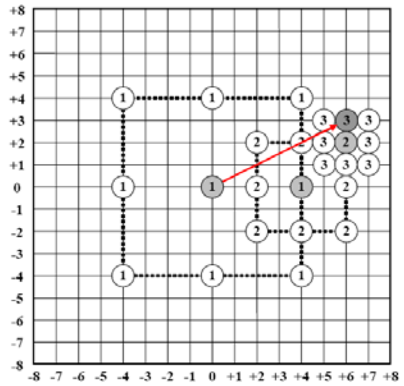


Figure 7. Illustration of three-step search

3.2. Three-Step Search

The three-step search was proposed by Koga et al. [13]. Figure 7 illustrates an example of the three-step search procedure. Like the 2D-log search, the process of three-step search also starts from the center point of the search area to find the best matching position of the eye template. Rather than examining five search points at each step as in the 2D-log search, it examines nine search points within the search area, including the center point and the other eight points with a square arrangement away from the center point with a step size, r, as shown in Figure 7. The initial value of r is equal to or slightly larger than half of the maximum search range ($r \geq d_{max}/2$) and is reduced to one half after each search step. The search procedure continues until step size reduces to one pixel, and the best matching point will be regarded as the best position of the eye. Similar to Figure 6, in the example of Figure 7 with $d_{max} = 8$, the three-step search requires 3 steps and 25 search points to reach the final destination.

4. Experimental Results

Experiments were performed to test the improved eye tracking component of the driver fatigue detection system. The same five driver videos used for testing our previous system were also used to verify this improved one for comparison. These videos were captured by a SONY PC115 color DV camera with 320×240 true color format. The first four videos were taken under different illumination conditions with different drivers and backgrounds. The fifth video was taken when driving around a parking lot at nightfall with large illumination change.

The testing environment was a personal computer with a Pentium D 3.20 GHz CPU and 1024 MB RAM. Table 2 lists the experimental results of eye tracking using the MAD matching function, with the exhaustive, 2D-log, and three-step search algorithms. Note that in these experiments, the maximum displacement parameters d_{xmax} and d_{ymax} both are set to 10 pixels for the search areas in eye tracking. Note also that the data of the exhaustive search with the MAD matching function were the eye tracking experimental results of our previous driver fatigue detection system.

Table 2. Results of eye tracking failure

Video No.	No. of Frames	Exhaustive search	2D-log search	Three-step search
1	2634	8	14	8
2	1524	6	14	12
3	2717	46	43	25
4	433	7	7	5
5	1399	280	200	191
Total	8707	347	278	241
Correct rate for Eye tracking		96.01%	96.81%	97.23%

As shown in Table 2, the total number of tracking failure in the exhaustive search is 347, which is higher than those, 278 and 241, respectively, in the 2D-log and three-step search.

Table 3 gives the numbers of search points required to reach the best matching point for each eye tracking with three different search algorithms. As can be seen from Table 3, the number of search points required for the exhaustive search is fixed, 441, due to $d_{x\_max} = d_{y\_max} = 10$. However, the average search points of 2D-log and three-steps are less than 24 and 33, which are about 5.38% and 7.48% of exhaustive search.

Table 3. Results of average search points

Video No.	Exhaustive search	2D-log search	Three-step search
1	441	22.90	32.99
2	441	22.79	32.97
3	441	23.33	32.99
4	441	24.90	32.96
5	441	24.73	32.98
Average	441	23.73	32.98
Ratio	100.00%	5.38%	7.48%

A typical eye template in our experiments is of size $w = 26$ and $h = 7$. Therefore, the number of pixels in the template is $n = 182$, and the number pixels in the corresponding search area is $m = 1242$ for $d_{max} = 10$. Table 4 lists the number of operations required for each search algorithm to find the optimal matching points with different matching function, based on the results of Table 3, where the exhaustive, 2D-log, and three-step searches are assumed to require 441, 24 (average upper bound), and 33 (maximum) search points, respectively. It is noted that the 2D-log search requires only up to 5.44% of computation of the exhaustive search.

Table 4. Operations required for each search

Operation Type	Exhaustive search	2D-log search	Three-step search
Addition	160,083	8,712	11,979
Absolute	80,262	4,368	6,006
Total	240,345	13,080	17,985
Ratio	100.00%	5.44%	7.48%

5. Conclusion

In this paper, we have presented an improved driver fatigue detection system over our previous scheme. Rather than using the exhaustive search in the previous system, we have applied fast search algorithms, such as the 2D-log search and the three-step search, in eye tracking to improve search efficiency. The experimental results have shown that the 2D-log search has the best performance. It needs only up to about 5.44% of computations required for the original scheme in eye tracking, while it can still reach a slight higher correct rate of tracking, 96.81%, as compared to the original scheme, 96.01%. This result makes our improved scheme more suitable to be implemented in embedded systems.

References

[1] T.C. Chieh, M.M. Mustafa, A. Hussain, E. Zahedi, and B.Y. Majlis, “Driver Fatigue Detection Using Steering Grip Force,” Proc. IEEE Student Conference on Research and Development, Putrajaya, Malaysia, pp. 45–48, Aug. 2003.

[2] K.J. Cho, B. Roy, S. Mascaro, and H.H. Asada, “A Vast DOF Robotic Car Seat Using SMA Actuators with a Matrix Drive System,” Proc. IEEE Robotics and Automation, New Orleans, LA, USA, vol.4, pp.3647–3652, Apr. 2004.

[3] W. Dong and X. Wu, “Driver Fatigue Detection Based on the Distance of Eyelid,” Proc. IEEE VLSI Design and Video Technology, Suzhou, China, pp. 365–368, May 2005.

[4] M. Eriksson and N.P. Papanikolopoulos, “Eye-Tracking for Detection of Driver Fatigue,” Proc. IEEE Intelligent Transportation Systems, Boston, MA, USA, pp. 314–319, Nov. 1997.

[5] R.C. Gonzalez and R.E. Woods, Digital Image Processing, Second Edition, Prentice Hall, Upper Saddle River, NJ, USA, 2002.

[6] H. Gu, Q. Ji, and Z. Zhu, “Active Facial Tracking for Fatigue Detection,” Proc. IEEE Workshop on Applications of Computer Vision, Orlando, FL, USA, pp. 137–142, Dec. 2002.

[7] H. Gu and Q. Ji, “An Automated Face Reader for Fatigue Detection,” Proc. IEEE Automatic Face and Gesture Recognition, Seoul, Korea, pp. 111–116, May 2004.

[8] H.M. Hang, Y.M. Chou, and S.C. Cheng, “Motion Estimation for Video Coding Standards,” J. VLSI Signal Process., vol. 17, pp.113–136, 1997.

[9] W.B. Horng and C.Y. Chen, “A Real-Time Driver Fatigue Detection System Based on Eye Tracking and Dynamic Template Matching.” Tamkang Journal of Science and Engineering, vol.11, no.1, pp.65–72, Mar. 2008.

[10] T. Ito, S. Mita, K. Kozuka, T. Nakano, and S. Yamamoto, “Driver Blink Measurement by the Motion Picture Processing and its Application to Drowsiness Detection,” Proc. IEEE Intelligent Transportation Systems, Singapore, pp. 168–173, Sept. 2002.

[11] J.R. Jain and A.K. Jain, “Displacement Measurement and Its Application in Interframe Image Coding,” IEEE Trans. Communications, vol.COM-29, no.12, pp.1799–1808, Dec. 1981.

[12] Q. Ji, Z. Zhu, and P. Lan, “Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue,” IEEE Trans. Veh. Tech., vol.53, no.4, pp.1052–1068, July 2004.

[13] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion-Compensated Interframe Coding for Video Conferencing,” Proc. IEEE Nat. Telecommun. Conf., New Orleans, LA, USA, pp.531–534, Nov. 1981.

[14] M.A. Recarte and L.M. Nunes, “Effects of Verbal and Spatial-Imagery Tasks on Eye Fixations while Driving,” J. Exp. Psychol. Appl., vol.6, no.1, pp.31–43, 2000.

[15] D. Shinar, Psychology on the Road, John Wiley & Sons, Danvers, MA, USA, 1979.

[16] S. Singh and N.P. Papanikolopoulos, “Monitoring Driver Fatigue Using Facial Analysis Techniques,” Proc. IEEE Intelligent Transportation Systems, Tokyo, Japan, pp. 314–318, Oct. 1999.

[17] Smart Motorist, Inc., “Driver Fatigue is an Important Cause of Road Crashes,” http://www.smartmotorist.com/traffic-and-safetyguideline/driver-fatigue-is-an-important-cause-of-road-crashes.html (visited, 2008/05/31).

[18] A. Vuckovic, D. Popovic, and V. Radivojevic, “Artificial Neural Network for Detecting Drowsiness from EEG Recordings,” Proc. IEEE Seminar on Neural Network Applications in Electrical Engineering, Belgrade, Yugoslavia, pp. 155–158, Sept. 2002.

[19] R. Wang R., K. Guo, S. Shi, and J. Chu, “A Monitoring Method of Driver Fatigue Behavior Based on Machine Vision,” Proc. IEEE Intelligent Vehicles Symposium, Columbus, Ohio, USA, pp. 110–113, June 2003.

[20] R. Wang, L. Guo, B. Tong, and L. Jin, “Monitoring Mouth Movement for Driver Fatigue or Distraction with One Camera,” Proc. IEEE Intelligent Transportation Systems, Washington, D.C., USA, pp. 314–319, Oct. 2004.

[21] B.J. Wilson and T.D. Bracewell, “Alertness Monitor Using Neural Networks for EEG Analysis,” Proc. IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing, Sydney, Australia, vol.2, pp.814–820, Dec. 2000.

[22] Z. Zhu and Q. Ji, “Real-Time and Non-intrusive Driver Fatigue Monitoring,” Proc. IEEE Intelligent Transportation Systems, Washington, D.C., USA, pp.657–662, Oct. 2004.