Real-Time Driver Gaze Direction Detection Using the 3D Triangle Model and Neural Networks

Wen-Chang Cheng
Dept. of Computer Science and Information Engineering
Chaoyang University of Technology
Taichung,Taiwan
wccehng@cyut.edu.tw

You-Song Xu
Dept. of Computer Science and Information Engineering
Chaoyang University of Technology
Taichung,Taiwan
abcd-meat@umail.hinet.net

***Abstract*—In this paper, we propose a real-time driver gaze direction detection system using a 3D triangle model and neural networks to monitor the driver's distraction. This method uses two cameras and open CV (Open Source Computer Vision Library) to locate the face, eyes and mouth position. The location of the eyes and mouth are automatically taken by two cameras. Then the information of their locations is transmitted to the triangle model, and the difference of stereo vision is calculated for depth information. Finally, we use neural networks as a classifier to accomplish our driver awareness system. The test data include 9 kinds of gaze directions, each containing 10 pictures. In the experiment result, we can achieve about an 83% detection rate base on our testing data with 9 directions. Through analyzing input face images, the model can run in real-time speed at about 30 frames per second on a normal computer.**

Keywords-Driver distraction; Face detection; Eye detection; Mouth detection; Neural networks; Stereo Vision.

I. INTRODUCTION

As the vehicle industry advances, car accidents are happening every day. In many studies, driver distraction and inattention are the main causes of automotive collisions. According to the report of the National Highway Traffic Safety Administration in the United States [1], most traffic accidents are caused by drivers' unsafe behavior. It can either be by using cell phones, falling asleep, or picking up objects while driving, etc. To avoid car accidents caused by the driver's lack of awareness, many studies [2, 3] have been conducted using visual information. When people are fatigued, there are some obvious changes in facial features such as eyelid blinking, head movement, or gaze direction [4]. Therefore, lots of image processing and classifier techniques are used in this issue.

Head pose estimation is intrinsically linked with visual gaze estimation. Physiological investigations have demonstrated that a person's prediction of gaze comes from a combination of both head pose and eye direction [5, 6]. There are also researches focusing on the high contrast between the sclera and the iris which is discernible from a distance and might contribute to facilitate gaze perception [7]. Consequently, human gaze estimation combines human motion [8], [9], face detection [10], face recognition [11], and affect recognition [12]. However, these methods in a certain degree detect the geometric changes in facial features to evaluate the driver's gaze direction. When we observe a driver's facial feature, we can find out that the change in the driver's gaze direction can transform the triangle model formed by the eyes and mouth. Therefore, through detecting the geometric changes in the triangle model, we can evaluate the driver's gaze direction.

Moreover, we find out that the driver's gaze direction even on the same horizontal height but turning left or right also has the same triangle model. However, it lacks depth information. In this paper, we make use of a real-time driver awareness system which not only detects the triangle model through the eyes and mouth, but also calculates the depth information through stereo vision. We call it as a 3D triangle model. Finally, we derive geometry feature vectors from the 3D-triangle model, and use a pre-trained neural network to do the classification. In the experiment result, this method can complete the real-time driver gaze detection correctly and effectively.

The remaining sections of this paper include: Section 2 which introduces the method flowchart, Section 3 which discusses the 3D-triangle model. Sections 4 and 5 show the neural networks and experimental results, as well as the last section which presents the conclusion.

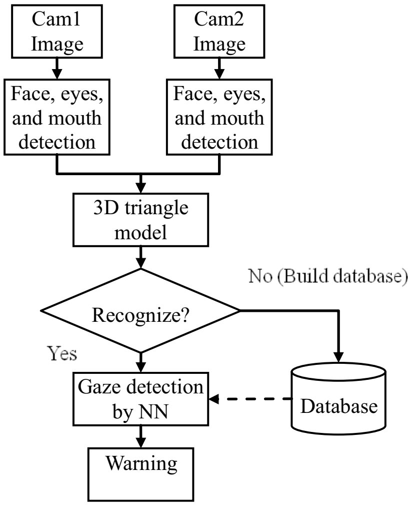


Figure 1. Flow chart of driver awareness system.

978-0-7695-5101-2/13 $26.00 © 2013 IEEE
DOI 10.1109/AMS.2013.11

II. OVERALL OPERATIONS OF THE SYSTEM

Figure 1 is the flow chart of the proposed method. When the system receives input images, the face detection function will be enabled. Then we can obtain the eyes and mouth positions to form the left and right triangle model, and depth information through the disparities of triangle models on Cam1 and Cam2s' images. On the left or right face image, we can get one feature vector which contains the triangle and depth information, named as the 3D-triangle model. After the following step, we have two choices, one is to perform recognition, and the other is to build a database for training neural networks. Finally, we can determine the gaze direction through trained neural networks. If the gaze direction is over a certain range, a warning message will be given.

III. 3D-TRIANGLE MODEL

In the beginning, the system has to locate the face, eyes, and mouth of the input image. We complete this problem through Viola and Jones' algorithm [13] and openCV (Open Source Computer Vision Library) function [14].

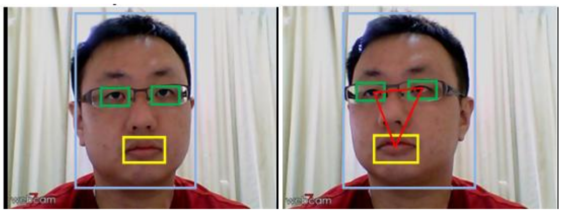
However, it is inefficient to locate the face, eyes, and mouth at the same time in the input image. It's possible to result in detection error under a complicated background. Therefore, we detect the face's location first, and then locate the eyes and mouth based on the face ROI (region of interest). It can increase detection accuracy and reduce the influence of a complicated background.

A. Face, Eyes and Mouth Detections

First we use function of openCV to detect the face's location. OpenCV is a library of programming functions mainly aimed at real-time computer vision, developed by Intel, and is now supported by Willow Garage and Itseez. It is free for use under the open source BSD license. The library is a cross-platform. It focuses mainly on real-time image processing. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate performance.

B. Triangle Model

As previously mentioned, detecting the eyes and mouth location through the whole input image is not only inefficient but can also result in detection error when under a complicated background. Therefore, we use the frame of the face to predict the location of the eyes and mouth. Then we detect the eyes and mouth through the pre-set ROI. It can improve the efficiency and accuracy of detection.



(a) (b)

Figure 2. Stereo triangle model, (a) left cam, and (b) right cam.

We obtain *x*-coordinate and *y*-coordinate of the left eye corner, right eye corner, and middle of the two lips through the openCV. In figure 2, we can see our detection result. The blue frame would detect the face's position. The green frames locate the eyes' position, and the yellow frame the mouth's position.

After we obtain the eyes and mouth positions, we set the left eye corner, right eye corner and middle of the two lips as three vertices. Three vertices can form an inverted triangle, so as our triangle model. When the face looks up, down, left, or right, the triangle breaks position. Therefore, we take the triangle's three angles and three sides as our features. By these features we can learn the gaze direction.

We can calculate the length of the three sides of a triangle by Euclidean distance. We set the left eye corner as apex '*A'*, right eye corner as apex '*B'*, and the middle of the two lips as apex '*C'*. After we have lengths *a*, *b* and *c* of $\overline{AB}$, $\overline{BC}$ and $\overline{AC}$ respectively (through Pythagorean theorem). We can calculate the three angles of the triangle by Law of cosines [15]. The law of cosines (also known as the cosine formula or cosine rule) relates the lengths of the sides of a plane triangle to the cosine of one of its angles. It is useful for computing the third side of a triangle when only two sides and their enclosed angle are known, and in computing the angles of a triangle if all three sides are known.

$$c^2 = a^2 + b^2 - 2ab\cos(\angle C) \tag{1}$$

$$b^2 = c^2 + a^2 - 2ca\cos(\angle B) \tag{2}$$

$$a^2 = b^2 + c^2 - 2bc\cos(\angle A) \tag{3}$$

Then every face image can transform to a triangle model.

C. 3D-Triangle Model Using Stereo Vision

In order to find the depth information of the left eye, right eye, and mouth, the light geometry rules have been used. The projection of the real world location onto the two image planes requires finding the exact location of the object [16]. In our system, we use two parallel cameras with a horizontal displacement as shown in Figure 3. The stereo configuration is derived from the pinhole camera model [17].

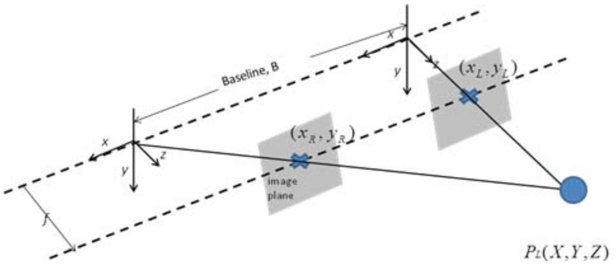


Figure 3. Stereo vision configuration

In Figure 3, (x_R, y_R) is the target camera center, and (x_L, y_L) is the reference camera center. The precondition of our system is based on the parallel cameras, which are moved along the x-coordinate. Consequently, y_R is equal to y_L. Moreover, *f* is the focal distance of the two cameras, and *B* is the baseline distance (distance between two cameras' center) about 10 cm. The points of the images can be described as the following:

$$(x_R, y_R) = (f\frac{X-B}{Z}, f\frac{Y}{Z}) \tag{4}$$

$$(x_L, y_L) = (f\frac{X}{Z}, f\frac{Y}{Z}) \tag{5}$$

The disparity *d* of the stereo images is the subtracting of two points, (x_L, y_L) and (x_R, y_R):

$$d = x_L - x_R = f\frac{X}{Z} - f\frac{X-B}{Z} \tag{6}$$

The location of the correct projections of the same point P_L on the two image planes can determine the depth of P_L in the real world. Therefore, we can define depth Z as:

$$Z = \frac{fB}{d} \tag{7}$$

As above mentioned, we have left eye, right eye, and mouth three points. We take Z_{AB}=Z_A-Z_B, Z_{BC}=Z_B-Z_C and Z_{CA}=Z_C-Z_A as our depth feature, where Z_A, Z_B, and Z_C are the depth of '*A*', '*B*', and '*C*' point of left triangle model respectively. Therefore our face model can be described as the following feature vector:

$$\mathbf{F} = [a, b, c, \angle A, \angle B, \angle C, Z_{AB}, Z_{BC}, Z_{CA}]^T, \tag{8}$$

IV. Gaze direction detection by NN

We use the back-propagation of neural networks as our training method. Back-propagation is a common method of training artificial neural networks so as to minimize the objective function. It is a supervised learning method and is a generalization of the delta rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks. The term is an abbreviation for "backward propagation of errors". Back-propagation requires that the activation function used by the artificial neurons (or "nodes") be differentiable [18].

In this paper, we take nine sets of data from a single face image as our input (training data). Output is our gaze direction. We use a tan-sigmoid transfer function in the first layer and pure-liner function in the second layer. In experiment, we train the data using different numbers of hidden layer nodes, and find out that we can reach a good quality result using about 20 nodes.

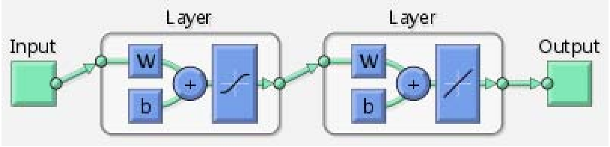


Figure 4. Architecture of used back-propagation neural networks.

V. Experimental results

The layout of our system is shown in Figure 5, with two cameras and one projection screen. The driver sits in front of a projection screen. Then we locate the driver's horizontal gaze on the screen by setting a datum point. Base on the datum point, we project a 3×3 area, therefore we have nine gaze directions including the middle. Then two cameras are used to obtain real-time face images which are processed through Viola and Jones' algorithm [13] and openCV function to obtain the eyes and mouth location. The cameras then capture the driver's face from two different angles; two different locations of the eyes and mouth can be combined to produce depth information.

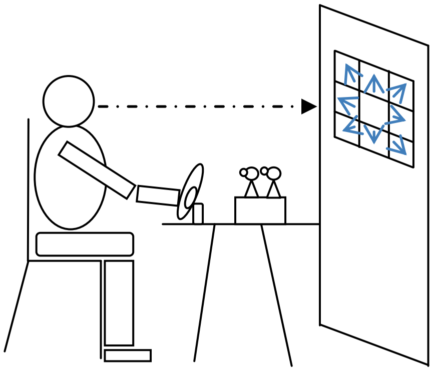


Figure 5. System design with stereo vision and projector

In our training process, we have nine gaze directions. In each gaze direction, we take a set of pictures with two IP cameras as shown in Figure 6. Then we extract the triangle model and depth information from each set of pictures. There are ten participators and each one has nine sets of pictures. Therefore, we have 90 face model vectors in 9 gaze directions. We randomly pick five face model vectors as the training data, and the other five people as the testing data, and repeat the experiments three times.

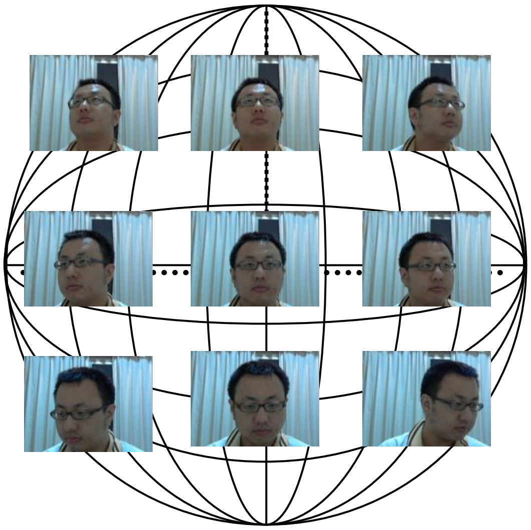


Figure 6. Gaze direction data sets.

We can see the training result as shown in Figure 7. As our figure, the red line suggests the results that we expected, and the blue line represents the training result. We can find out that the trend of the blue line generally fits the true result.

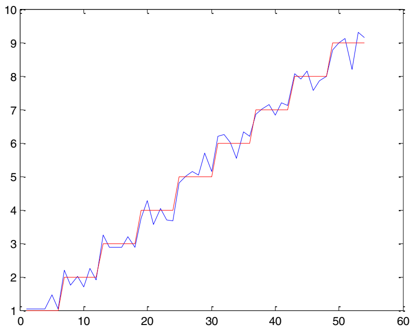


Figure 7. Training result

Consequently, we can obtain our experiment results in Table 1, and have an average of 81% detection rate.

TABLE I. Detection Rate

State	Correct	Error
Experiment 1	79%	21%
Experiment 2	84%	16%
Experiment 3	81 %	19 %

We also consider the number of hidden layer nodes in our experiment. When we train the data, we use different numbers of nodes. In each number, we train the data for ten times and we average the mean square error that shows that about 20 nodes can obtain better efficiency.

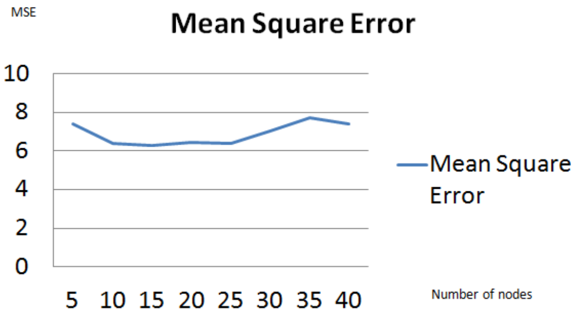


Figure 8. MSE with different numbers of nodes.

VI. Conclusions

In this paper, we propose a real-time driver gaze detection using neural networks with the 3D-triangle model features, face, eyes and mouth detection using the openCV functions. Back-propagation of neural networks is used for classification and training. In the experiment result, we can achieve about 83% detection rate based on our testing data with 9 directions. However, the proposed method still has some miss detection problems. It may cause detecting error when the head is moving too fast or is out of range. In the future, we are considering to use a tracking technique to increase the detection rate.

Acknowledgment

We show our gratitude to the Program Institute of National Science Council, (with number NSC-99-2632-E-324-001-MY3) for their support and help in obtaining the results of this paper

References

[1]. NHTSA, "Drowsy drivers detection and warning system for commercial vehicle drivers: Field proportional test design, analysis, and progress" National Highway Traffic Safety Administration, Washington, DC, http://www.nhtsa.gov/

[2]. Sung Joo Lee, Jaeik Jo, Ho Gi Jung, Kang Ryoung Park, and Jaihie Kim, "Real-Time Gaze Estimator Based on Driver's Head Orientation for Forward Collision Warning System," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 1, 2011, pp 254-267

[3]. Erik Murphy-Chutorian, and Mohan Manubhai Trivedi, "Head Pose Estimation in Computer Vision: A Survey," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, 2009, pp. 607-626.

[4]. Erik Murphy-Chutorian, and Mohan Manubhai Trivedi, "Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness," IEEE Trans. on *Intelligent Transportation Systems*, VOL. 11, NO. 2, 2010, pp. 300-311.

[5]. V.F. Ferrario, C. Sforza, G. Serrao, G. Grassi, and E. Mossi, "Active Range of Motion of the Head and Cervical Spine: A Three-Dimensional Investigation in Healthy Young Adults," J. Orthopaedic Research, vol. 20, no. 1, pp. 122-129, 2002.

[6]. S. Langton, H. Honeyman, and E. Tessler, "The Influence of Head Contour and Nose Angle on the Perception of Eye-Gaze Direction," Perception and Psychophysics, vol. 66, no. 5, pp. 752-771, 2004.

[7]. H. Kobayasi and S. Kohshima, "Unique Morphology of the Human Eye," Nature, vol. 387, no. 6635, pp. 767-768, 1997.

[8]. T. Moeslund, A. Hilton, and V. Kru¨ ger, "A Survey of Computer

Vision-Based Human Motion Capture," Computer Vision and Image Understanding, vol. 81, no. 3, pp. 231-268, 2001.

[9]. T. Moeslund, A. Hilton, and V. Kru¨ ger, "A Survey of Advances in Vision-Based Human Motion Capture and Analysis," Computer Vision and Image Understanding, vol. 104, no. 2, pp. 90-126, 2006.

[10]. E. Hjelma°s and B. Low, "Face Detection: A Survey," ComputerVision and Image Understanding, vol. 83, no. 3, pp. 236-274, 2001.

[11]. W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," ACM Computing Surveys, vol. 35, no. 4, pp. 399-458, 2003.

[12]. B. Fasel and J. Luettin, "Automatic Facial Expression Analysis: A Survey," Pattern Recognition, vol. 36, no. 1, pp. 259-275, 2003.

[13]. Paul Viola and Michael J. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features," IEEE CVPR, 2001.

[14]. OpenCV, http://sourceforge.net/projects/opencvlibrary/

[15]. Law of cosines: http://en.wikipedia.org/wiki/ Law_of_cosines.

[16]. A. Bovik, Handbook of Image and Video Processing. Elsevier Academic Press, 2005.

[17]. Y. Morvan, "Acquisition, Compression and Rendering of Depth and Texture for Multi-view Video." Thesis PhD. Eindhoven University of Technology, 2009.

[18]. Satish Kumar, "Neural Networks: A Classroom Approach ," International Edition 2005, Exclusive rights by McGraw-Hill Education (Asia).

[19]. Nurulfajar Abd Manap, Gaetano Di Caterina, John Soraghan, Vijay Sidharth, and Hui Yao, "Smart Surveillance System Based on Stereo Matching Algorithms with IP and PTZ Cameras," CeSIP, Electronic and Electrical Engineering, University of Strathclyde, UK.