Eye-Motion Detection System for MND Patients

Chu-Lian Xu and Chyi-Yeu Lin
Department of Mechanical Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
xuchulian@gmail.com; jerrylin@mail.ntust.edu.tw

***Abstract*—This paper aims to develop an eye-motion based communication system for motor neuron disease (MND) patients to contact with care providers any time they want when they lie on the bed. This eye-motion detection system involves technical modules of eye-blink detection, gaze estimation and head pose estimation on MND patients. The system comprises a rotating arm with a camera, an infrared light source and a speaker. The arm with the camera will autonomously rotate to track the face of the patient with arbitrary body directions so that necessary eye detections can be properly conducted when it is called. When the patient needs to call out, only designated blinking and eye motions will trigger the call out action for the need of finding the care provider.**

Keywords-eye-motion detection; motor neuron disease; blink detection; gaze estimation; head pose estimation

I. INTRODUCTION

Motor neuron disease (MND) is a rare but devastating illness which leads to progressive paralysis and eventual death [1]. During later periods, MND patients lose mobility due to organ failures. The only manageable body parts of most MND patients in their final stages are their eyes and brains, which make them difficult to communicate with others. Because they lose the ability to speak and write, they can only contact the outside world through human-computer interaction; e.g. controlling brain waves or tracking eye movements. Currently, brain wave controlling devices need to be worn by users, so they are not convenient for people to use. There exists eye-motion based software which enables the MND patients to write in the computer by using their eye functions only. When they are away from PC and lie on the bed, they cannot communicate with care providers. With the goal of helping MND patients on the bed to call for other people with a simple and easy approach, this research aims to develop an eye-motion detection system, which can successfully detect the eye motions regardless the head directions, day or night. This system incorporates different visual technologies, such as face detection, eye center localization, head-pose estimation, and feature point extraction. Fig. 1 shows the schematic diagram of the system. The patient lies on the bed. The automatic face-tracking device is installed to the bed board. An IR camera is installed on the device and points to user's face.

II. VISUAL AND TRACKING SYSTEMS

Blink detection needs to focus on eyes. Observing the feature points in the eye area is the most direct way. Therefore, eye tracking and localization are essential to the system. Firstly, the face of the patient should be contained in the image and detected. The basic work of the image processing is to detect the face in the image, then locate the eyes on the face, and finally locate the eye centers. Several popular methods for face detection have been developed, such as Haar Feature-based Cascade Classifier [2], HD face detection of Kinect v2 SDK, and Facial landmarks detection with Dlib [3]. This research adopts the facial landmarks detection with Dlib. There are 4 popular eye-center localization methods, including projection method [4][5], Hough transform method [6], AdaBoost classifier method [7], and template matching method.

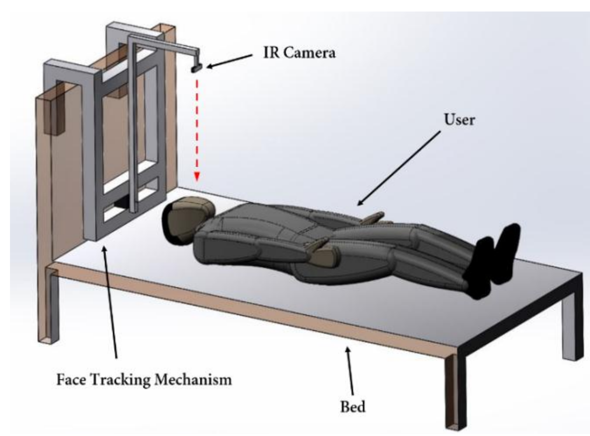

Figure 1. Schematic diagram of the system.

A. Face Detection

Dlib is an open source C++ library containing machine learning algorithms and tools for creating complex software. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, and mobile phones, and it works on Windows, Linux, and OSX. In recent years, much of the development has been focused on creating a broad set of statistical machine learning tools [3]. Fig. 2 demonstrates the results of Dlib face detection. The facial landmark detector implemented inside Dlib produces 68 coordinates that map to specific facial structures.

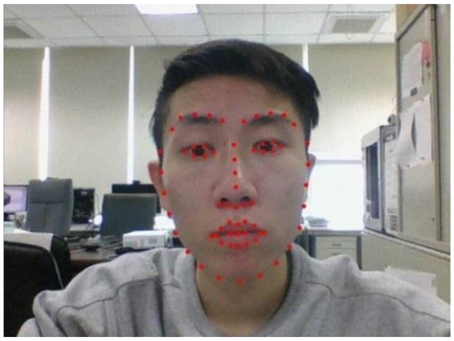

Figure 2. Example of face detection using Dlib.

B. Eye Centre Localization

Our method uses a gradient-based algorithm by Timm and Barth to locate eye centers [8]. It uses the inner product of the displacement vector and the gradient vector to localize the eye. In Equation (1) and (2), $\mathbf{c}$ represents the possible center, $\boldsymbol{d_i}$ the unit vector of displacement from every pixel to the possible center, and $\mathbf{g_i}$ the gradient vector of each pixel. Fig. 3 shows that when two vectors get closer to parallel relationships, the inner product value increases. The maximum value calculated is likely to be the position of the eye center.

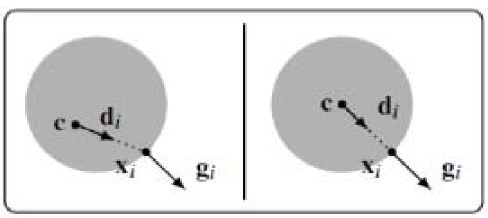

Figure 3. Normalized displacement vector $\boldsymbol{d_i}$ and gradient $\boldsymbol{g_i}$ [9].

$$\boldsymbol{c}^* = \underset{c}{\arg max} \left\{ \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{d}_i^T \boldsymbol{g}_i)^2 \right\} \quad (1)$$

$$\boldsymbol{d}_i = \frac{\boldsymbol{x}_i - \boldsymbol{c}}{\|\boldsymbol{x}_i - \boldsymbol{c}\|_2} \quad , \quad \forall \mathrm{i}: \|\boldsymbol{g}_i\|_2 = 1 \quad (2)$$

III. EYE-MOTION DETECTION SYSTEM

A. Hardware

1) Infrared (IR) Camera

Since the eye-motion based communication system for MND patients will mainly be installed on the bed to assist them communicate with others during both daytime and night time, an IR camera are selected, as shown in Fig. 4. This functioning of this IR camera is similar to a webcam and it has a USB portal. There are 24 infrared lamps installed around the camera lens to provide adequate lighting at night. Besides, the camera has dual IR-CUT filters. When infrared induction points sense the changes of light, the built-in IR-CUT automatically shifts filters. In other words, it can obtain the optimum imaging effects in the daytime or at night.

Figure 4. Example of IR camera with 24 infrared lamps.

2) Face Tracking Mechanism

Patients on the bed are often turned by care providers to change their body direction so as to reduce the pain related to excessive hours of lying with the same posture. Their face directions will also change when care providers change their postures. Therefore, we need a system which can track the face of a patient and automatically move the IR camera to view the face in the normal direction any time. This mechanism comprises an IR camera-supporting arm which can be rotated so that the camera is able to move along the circumference of a semicircle with the face as the center. The image-based visual servoing technique is implemented to perform the tracking motion [9]. The rotation segment of the face tracking mechanism consists of a stepping motor, Arduino UNO, a stepping driver, and a power supplier.

B. Mathematical Model

1) Blink Detection

After the camera obtains the image of the face of the patient, we can begin the image processing specifically for ROI. As shown in Fig. 5, the left eye and the right eye are extracted, as illustrated in the left bottom and the right bottom.

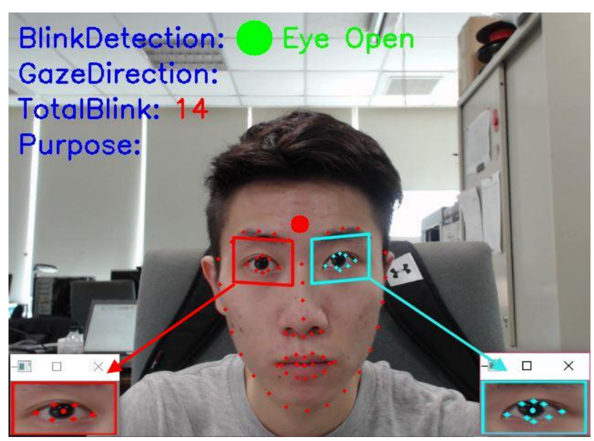

Figure 5. Eye image segmentation.

To build the blink detector, we implemented the method that primarily calculates the length-width ratio and then sets a threshold value to detect the blinks [10]. As shown in the left image of Fig. 6, after we have detected the eight eye feature points, P_1 to P_8, the EAR (Eye Aspect Ratio) can be calculated by Equation (3). The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks. The denominator is proportionally weighted since there is only one set of horizontal points but three sets of vertical points. In this approach, the eye aspect ratio is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place.

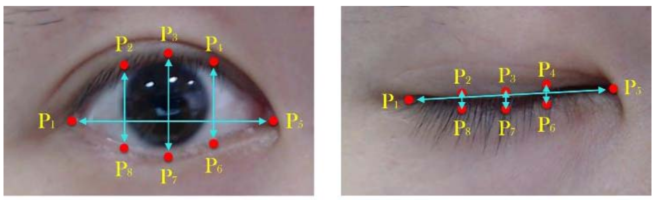

Figure 6. A visualization of eye landmarks when the eye is open (left). Eye landmarks when the eye is closed (right).

$$EAR = \frac{\|p_2-p_8\|+\|p_3-p_7\|+\|p_4-p_6\|}{3\|p_1-p_5\|} \tag{3}$$

2) Head Pose Estimation

The previous section mentioned that the face of the patient may be constantly turned to a different direction. With the goal of autonomously moving the camera to the direction normally facing to the face of the patient, we need to develop an image processing code to detect the angle between the camera and the face. The method we used is based on 3D-2D point correspondence and then fit the points to the 3D model. OpenCV provides a method (solvePnP) that can handle this task. To calculate the 3D pose of an object in an image we need two types of basic information. Firstly, we need the 2D (x, y) locations of a few points in the image of the face. In our case, we choose the corners of the eyes, the tip of the nose, corners of the mouth and the chin. Dlib's facial landmark detector provides us with many points to be chosen from. Secondly, we need the 3D location of the 2D feature points. For this, we downloaded a free model of a human head from the net, and used MeshLab to mark some feature points on the model (see Fig. 7).

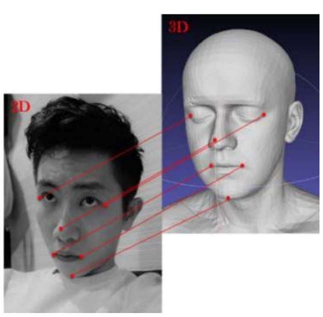

Figure 7. Example of 3D–2D point correspondence.

3) Perspective transformation:

After detecting the face and eye feature points, eye positioning and separation are preliminarily completed. Since the separation of eyes depends on the surrounding feature points, when we turn our head to the left or right, the eye images will also be inclined or even distort. As a result, the abnormal images may ill-impact the detection. Hence, we need to adopt the perspective transformation (PT). PT projects the image onto a new viewing plane, which is also known as projective mapping. The common transformation formula is:

$$[x', \quad y', \quad w'] = [u, \quad v, \quad w]\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{4}$$

where u and v are the coordinates on the original image. They can be transformed to the coordinates x, y, after the relevant conversion, $x = x' / w'$ and $y = y' / w'$. The transformation matrix on the right of Equation (4) can be divided into four parts. The first part is an internal matrix, consisting of a_{11}, a_{12}, a_{21}, a_{22}. It represents linear variations, such as scaling, shearing and rotation, where $[a_{31} \quad a_{32}]$ represents the translation. The matrix $[a_{31} \quad a_{32}]^T$ aims to induce the perspective transformation. Hence, affinity is a special form of the perspective transformation. Fig. 8 shows implementation of perspective transformation.

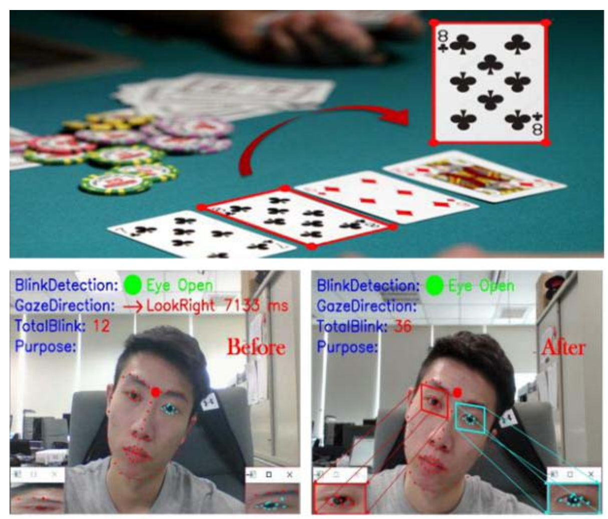

Figure 8. Example of perspective transformation.

4) Blink Detection and Gaze estimation:

As shown on Fig. 6, when the eye is fully opened, the eye aspect ratio would be consistently a large value. However, once the user blinks, the eye aspect ratio decreases quickly to near zero when it is fully closed. On the other hand, the eye center and eye feature points are measured referenced to the affine of the eye movement and projection area. Base on the gradient-based algorithm, the eye gaze screen coordinates mapping can be obtained as shown in Fig. 9.

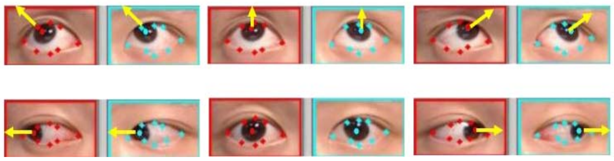

Figure 9. Detection of the eye center and eye feature points.

5) Head Pose Estimation:

In this research, we used the corners of the eyes, the tip of the nose, corners of the mouth and the chin as the 3D facial feature points on a 3D human head model.

Note that the above 3D feature points are in some arbitrary reference frame system, also referred to the world coordinates. Fig. 10 demonstrates the results of the head pose estimation.

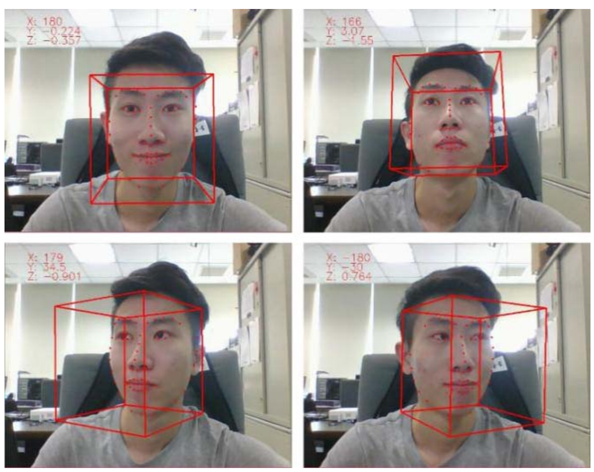

Figure 10. Examples of head pose estimation.

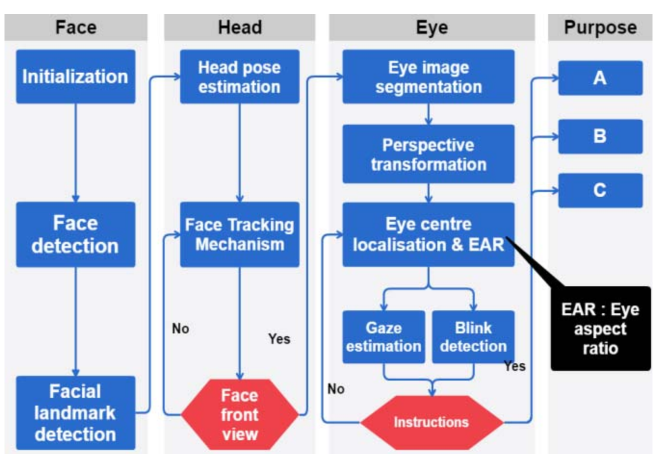

Figure 11. Flowchart of the communication system.

C. Detection System

The operation of the eye-motion based detection system consists of 11 steps, as shown in Fig. 11. The content of these steps are described below.

1) Initialization: Connect and set up hardware and arrange the setting according to different environments.

2) Face detection: The face detection is executed so as to locate the entire face.

3) Facial landmark detection: The pre-trained facial landmark detector inside the Dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face.

4) Head pose estimation: The function (solvePnP) in OpenCV is used to estimate the pose.

5) Face tracking mechanism: Trigger the hardware and arrange the setting according to different environments.

6) Eye image segmentation: When the face image is obtained, only the face region will be used for the search of eyes. It improves both accuracy and speed.

7) Perspective transformation: Implement the perspective transformation to segment the eye image to eliminate the errors resulted from the face movement.

8) Eye centre localization and EAR: The center of the eye can be detected by analyzing the vector field of image gradients. Equation (3) is used to caculate the eye aspect ratio.

9) Gaze estimation: The eye center and eye feature points are measured with respect to the affine of the pupil movement and the projection area..

10) Blink detection: Based on the ratio of the eye landmark distances, blinking can be detected.

11) Instructions: Combine the gaze estimation and blink detection of the user to determine the intention of the patient. This part is currently under investigation.

IV. Experiments and Results

A. Experiments

In the experiment, the subject lies on the bed. His head faces upward. The IR Camera is set upon the rotation axis right above the user. It is also right across the user’s face and about 50 cm away. The tracking mechanism can autonomously rotate the camera-supporting arm so that the IR camera can always view the face in the normal direction, as shown in Fig. 12.

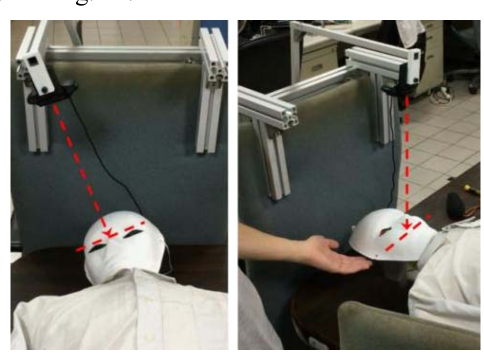

Figure 12. Demonstration of the face tracking mechanism.

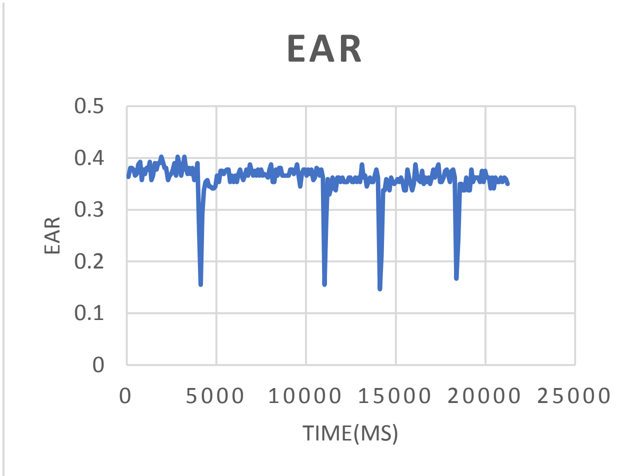

Figure 13. EAR history over a 20-second time.

B. *Statistical Results*

1) *Blink detection with different distances*

In this part of testing, the accuracy of blink detection was tested at five different distances: 50cm, 80cm, 100cm, 120cm and 140cm between the camera and the face. Subjects blinked eyes for 50 times at each of these distances. The results are shown in Table I. In cases that the distance between the camera and the face is less or equal to 120 cm, the accuracy of the blink detection is 100%.

TABLE I. RESULTS OF BLINK DETECTION

	Detected Blinks by the System	**Times of Blink**	**Accuracy**
50cm	50	50	100%
80cm	50	50	100%
100cm	50	50	100%
120cm	50	50	100%
140cm	49	50	98%

2) *Relationship between EAR and time:*

In this part of testing, the subject kept a distance of 50 cm from the camera. A 20-second-long footage of subject-blinking was extracted from the image terminal. The EAR values along the 20 seconds are plotted in Fig. 13, where x-axis represents time (ms), and y-axis EAR. The chart shows that the EAR value fluctuates between two major zones, one larger than 0.3 when the eyes are opened and the other below 0.2 when eyes approach closed. Every crossing between two zones, a dip in the figure, means a blink of the subject.

V. CONCLUSIONS

This research successfully develops a low-cost, easy-to-use and effective eye-motion detection system for MND patients. This eye-motion detection system can track the face of the patient in an autonomous manner. By that, the system is robust against unavoidable or unintentional head movement. The IR-camera based system enables the system to detect the patient in both bright and dark environment. Experiment results show that this system is highly accurate in eye-motion detection.

The eye-motion based communication system for MND patients can be further constructed, after the protocol of determining the intention of communication and the content of the communication is constructed. This protocol can only be based on the eye-motions, such as the number of blink, the gaze direction, and the time duration when a specific gaze direction is kept. After the system realizes that the patient wishes to call out, the rest will be the content of communication, such as “I need help”, “Please bring me watch”, or “I want to get up’. The classification of the content can be implemented by also specific eye motions. Such a communication system is currently under investigation. The corresponding results will be reported soon.

REFERENCES

[1] C. J. McDermott and P. J. Shaw, “Diagnosis and Management of Motor Neurone Disease,” BMJ, vol. 336(7645), pp. 658–662, Mar 2008.

[2] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” CVPR, Dec 2001.

[3] D. E. King, “Dlib-ml: A Machine Learning Toolkit,” JMLR, vol. 10, pp. 1755–1758, Jul 2009.

[4] Z. H. Zhou and X. Geng, “Projection Functions for Eye Detection,” Pattern Recognition, vol. 37(5), pp. 1049–1056, May 2004.

[5] Y. Zheng and Z. F. Wang, “Minial Neighborhood Mean Projection Function and Its Application to Eye Location,” J Softw, vol. 19(9), pp. 2322–2328, Sep 2008.

[6] N. Cherabit, F. Z. Chelali and A. Djeradi, “Circular Hough Transform for Iris Localization,” Science and Technology, vol. 2(5), pp. 114–121, Sep 2012.

[7] S. Z. Li and Z. Q. Zhang, “FloatBoost Learning and Statistical Face Detection,” IEEE Trans Pattern Anal Mach Intell, vol. 26(9), pp. 1112–1123, Sep 2004.

[8] F. Timm and E. Barth, “Accurate Eye Centre Localisation by Means of Gradients,” VISAPP, vol. 1, pp. 125–130, Mar 2011.

[9] C. Y. Lin, L. T. Son, Y. L. Chang, and Y. S. Shiue, “Image-Sensor-Based Fast Industrial-Robot Positioning System for Assembly Implementation,” Sensors and Materials, vol. 29(7), pp. 935-945, 2017.

[10] T. Soukupová and J. Čech, “Real-Time Eye Blink Detection using Facial Landmarks,” Computer Vison Winter Workshop, Feb 2016.