SUGISAKA MASANORI

Journal of Advances in Artificial Life Robotics
Vol. 1(1); June (2020), pp. 40–45
ON LINE ISSN 2435-8061; ISSN-L 2435-8061
https://alife-robotics.org/jallr.html

Research Article

Mouse Cursor Control System using Facial Movements

Masayoshi Tabuse[1], Manase Mizobe[2], Yasunari Yoshitomi[1], Taro Asada[1]
[1]*Graduate School of Life and Environmental Sciences, Kyoto Prefectural University, 1-5 Nakaragi-cho, Shimogamo, Sakyo-ku, Kyoto 606-8522, Japan*
[2]*Toray Engineering Co., Ltd., 3-22 Yaesu Ryumeikan Building, Yaesu 1-chome, Chuo-ku, Tokyo 103-0028, Japan*

ARTICLE INFO

Article History

Received 13 November 2019
Accepted 16 July 2020

Keywords

Mouse cursor system
Dlib C++ library
Facial movements
Physically disabled people

ABSTRACT

It is necessary to support of computer operation for a physically disabled person. One of the possible physical movements of the physically disabled person is facial movement. Recognition of facial movement of a person makes it possible to operate a computer. Furthermore without the adjustment for a user and adjustment for the distance from a user, it is possible to reduce the burden on a user. We developed a system to resolve these problems. In our system, a web camera, dlib C++ library and OpenCV library are used to extract feature points of the face and obtain the face direction. Changing the face direction, we can move a mouse cursor. Recognizing an open mouth or closed eye, we can carry out an operation of mouse click. In this paper, we evaluated the effect on operability due to the face direction and recognition rate due to distance.

© 2022 *The Author*. Published by Sugisaka Masanori at ALife Robotics Corporation Ltd.
This is an open access article distributed under the CC BY-NC 4.0 license
(http://creativecommons.org/licenses/by-nc/4.0/).

1. Introduction

Physically disabled people cannot move their limbs freely. Therefore, it is difficult for a physically disabled person to use a computer. The importance of the computer in the information society is increasing in a current social background. Therefore, support of the computer operation for physically disabled people is required. The interface for physically disabled people has been developed so that they can use a computer recently. There are two types of the interface. One is the contact-type which operates while attaching the device to a body. Another is the noncontact-type which operates by recognizing the movement of a body. However these interfaces are very expensive. Furthermore, although the contact-type has the advantage of easy detection of body movement, the user must attach it directly during use. On the other hand, noncontact-type must adjust parameters for a user. To resolve these problems reduces a burden for the user.

One of the possible physical movements of a physically disabled person is facial movement. This means the face direction and open and closed mouth or eyes. If a computer is operated with recognizing the facial movement, the physically disabled person can use it. We developed a system which operates a computer by the facial movement. In the previous study[1], we proposed a mouse cursor control system using Kinect. However Kinect got out of production. Therefore we developed a mouse cursor control system using a web camera instead of Kinect as a low cost interface. Our system consists of a computer, a web camera, dlib C++ library[2] and OpenCV library[3]. It extracts feature points of the face and estimates the face direction. We can move a mouse cursor by changing the face direction. We can carry out an operation of mouse click by recognizing an open mouth or closed eye. Our system is a noncontact-type interface. Therefore, the user doesn't have to attach the device to a body as the contact-type. Furthermore our system doesn't have to adjust parameters for each use. These reduce the burden on the user.

2. Proposed System

2.1. *System overview*

In our system, a computer, a web camera, dlib C++ library and OpenCV library are used to extract feature points of the face and estimate the face direction. Dlib is an open source C++ library containing image processing

Corresponding author's E-mail: tabuse@kpu.ac.jp

and machine learning algorisms. Dlib can extract a face area in image and detect 68 facial feature points from the face area. OpenCV estimates a face direction using obtained facial feature points. Our system has three functions for a physically disabled person. Firstly, it can control a mouse cursor depending on face direction. Secondly, it can perform a mouse click by recognizing an open mouth. Finally, it can carry out a mouse click by judging the open or closed eyes. In the following subsections, these functions are explained in detail.

2.2 *Control of a mouse cursor depending on the face direction*

Dlib extracts a face area in image and detects 68 facial feature points from the face area. We use 2 dimensional locations of 6 facial feature points (nose tip, chin, left eye left corner, right eye right corner, left mouth corner and right mouth corner) in the image to estimate a face direction. The solvePnP function in OpenCV estimates the rotation vector (the axis of rotation and the angle of rotation) of the face using these 6 locations, 3 dimensional face model and intrinsic camera parameters. The Rodrigues function in OpenCV converts the rotation vector to a rotation matrix R. $pitch$ and yaw which express a value of the angle for the vertical and horizontal face directions are calculated as follows:

$$pitch = \tan^{-1}(R_{32} / R_{33})$$
$$yaw = \tan^{-1}(R_{21} / R_{11}) ,$$

where $R = (R_{ij})$ is a rotation matrix.

We define the face direction vector $\boldsymbol{a}$ as

$$\boldsymbol{a} = (yaw,\ pitch) .$$

The control of a mouse cursor is operated according to the face direction vector $\boldsymbol{a}$. The mouse cursor is moved in the same direction as the vector. The moving speed of a mouse cursor is changed according to the size of the vector. When $|\boldsymbol{a}| < 10$, we define the face is facing to the front and the mouse cursor does not move. When $10 \leqq |\boldsymbol{a}| < 15$, the mouse cursor moves slowly. When $15 \leqq |\boldsymbol{a}| < 20$, the mouse cursor moves normally. When $20 \leqq |\boldsymbol{a}|$, the mouse cursor moves fast.

2.3. *Recognition of open mouth*

We extract four feature points of mouth (top, bottom, left and right) using dlib, as shown in Fig. 1. The vertical length of the mouth is found by feature point coordinates of the top and bottom. The horizontal length of the mouth is found by feature point coordinates of the left and right. The rate of the open mouth R_m is expressed as $R_m = h_m / w_m$ where h_m and w_m denote the vertical and horizontal length of the mouth, respectively. Let Th_m be the threshold for judgment of open or closed mouth. When $R_m > Th_m$, it is recognized that the mouth is opened. In the present study, the value of Th_m is set to 0.45, which was determined experimentally.

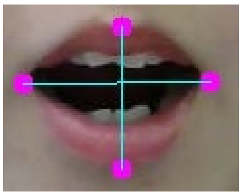


Fig. 1. Feature points (magenta circle) of mouth extracted using dlib and the vertical and horizontal length.

2.4. *Judgment of open and closed eyes*

We extract four feature points of eyes (top, bottom, left and right) using dlib, as shown in Fig. 2. The range of x-axis is the difference of x-coordinate between the left and right, and the range of y-axis is the difference of y-coordinate between the top and bottom. We define these ranges as eye regions shown as two white rectangles in Fig. 2.

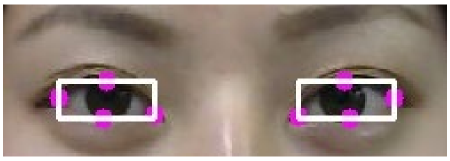


Fig. 2. Feature points (magenta circle) of eye extracted using dlib and eye regions (white rectangle).

The judgment of open or closed eye uses binary images of eye regions. For the binarization, at first, the extraction of the eye region is performed in a RGB image. It is converted from the RGB image to a grayscale image. Then we binarize the grayscale image using the discriminant analysis method[4]. Then the vertical and horizontal lengths of the eye are determined in the binary image, as shown in Fig. 3. We make the histogram of the number of black pixels in the vertical direction. The vertical length of the eye is the maximum on the histogram, and the horizontal length of the eye is the range of x-coordinate in the eye region. The rate of the open eye R_e is given by $R_e = h_e / w_e$ where h_e and w_e denote the vertical and horizontal length of the eye, respectively. Let Th_e be the threshold for judgment of open or closed eye. When $R_e > Th_e$, the eye is judged to be opening state. When $R_e \leqq Th_e$, the eye is judged to be closing state. In the present study, the value of Th_e is set to 0.3, which was determined experimentally.

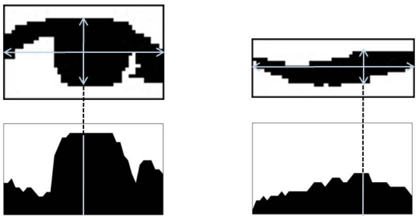


Fig. 3. The vertical and horizontal length and histogram of open eye (left) and closed eye (right).

2.5. *Mouse click processing*

Mouse click processing is performed by recognizing the intentional movement of the user's mouth and eyes. The judgment of intentional movement is the following conditions:

- the duration time of opening mouth state is *t1* or more,
- the duration time of closing eye state is *t2* or more,

where *t1* and *t2* are arbitrary times. In the present study, the value of *t1* and *t2* are set to 1.0 s and 0.8 s, respectively, which was determined experimentally. Mouse click processing is carried out only in the case of $|\boldsymbol{a}| < 10$.

3. Experiments

3.1. *Conditions*

The experiment was performed in the following computational environment: the PC was a HP ENVY 700-560jp (CPU: Intel(R) Core(TM) i7-4790 CPU 3.60GHz, memory: 8.00GB); the OS was Microsoft Windows 8.1 Pro; the development language was Microsoft Visual C++ 2015 Express Edition and the libraries were dlib 19.16 and OpenCV 3.4.1. The image was produced by Logicool C270 HD Webcam which was placed on the top of the display, as shown in Fig. 4.

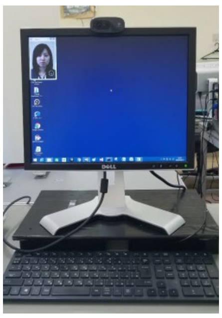
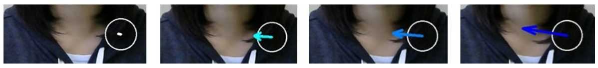


Fig. 4. Position of PC and a web camera in experiments.

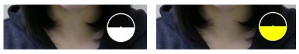
3.2. *User interface*

The user's face image is displayed on the upper left of the screen during mouse cursor operation and an arrow and a circle are displayed in the face image in order to the mouse cursor operation visually understandable. The arrow represents the face direction vector ***a*** shown in Fig. 5. The radius of the circle indicates $|\boldsymbol{a}|=10$. The lower half circle and the upper half circle represent the open mouth state and the closed eye state shown in Fig. 6 and 7, respectively.



(a) (b) (c) (d)

Fig. 5. The arrow represents the face direction vector ***a***. (a) $|\boldsymbol{a}| < 10$, (b) $10 \leqq |\boldsymbol{a}| < 15$, (c) $15 \leqq |\boldsymbol{a}| < 20$, (d) $20 \leqq |\boldsymbol{a}|$.



(a) (b)

Fig. 6. The open mouth state. (a) the judgment of the open mouth, (b) the mouse click.

(a) (b)

Fig. 7. The closed eye state. (a) the judgment of the closed eye, (b) the mouse click.

3.3. *Method*

We conducted three kinds of experiments which were performed by six subjects in their twenties (2 males without glasses, 2 females without glasses and 2 females with glasses). For the first experiment, we measured times that a mouse cursor moved depending on the face direction from a circle drawn on screen to another circle drawn in a distance of 20 cm, shown in Fig. 8. The subject sat at a distance of 0.8m from a web camera and moved the mouse cursor to eight directions (vertical, horizontal, and diagonal). This experiment was conducted to assess the operability by the difference of directions. For the second experiment, we examined the recognition rate of mouse click by open mouth. The distance between the web camera and the subjects was conducted at 0.6m, 0.8m, 1.0m, 1.5m, and 2.0m for evaluating influence to the recognition rate by the difference in distances. The subjects performed three times of the action of opening the mouth for one second or more. The recognition rate is the actual mouse click rate. For the last experiment, we conducted experiment to judge open or closed eyes. This experiment was conducted at distance of 0.6m, 0.8m, 1.0m, 1.5m, and 2.0m between the web camera and subjects for evaluating influence to the judgment accuracy by the difference in distance. We evaluate the accuracy of judgment with F-measure. F-measure is expressed the following equation:

$$F = \frac{2PR}{P+R},$$

where P, R and F denote the rate of frames which is really closing eye among frames judged as closed state, the rate of frames judged as closing state among frames which is closing eye, and the harmonic mean of precision rate P and recall rate R, respectively. We performed judgment of closed eye by visual observation.

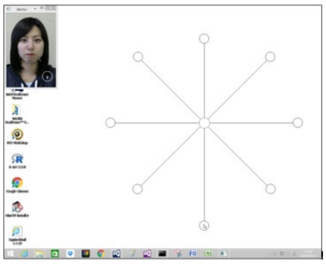


Fig. 8. The experiment of mouse cursor movements.

3.4. *Results and discussion*

4.3.1. Movement of mouse cursor depending on the face direction

Fig. 9 shows the time to move the mouse cursor from a circle to another circle with 6 subjects. From the times to move of 6 subjects, it was the movement of top and bottom directions to take longer time than other directions. The subjects B and C wearing glasses took more time to move in all directions other than the bottom left compared to subjects without glasses. It is probably because the subject's glasses frame was misrecognized as subject's eye and this misrecognition interrupt subject's operation.

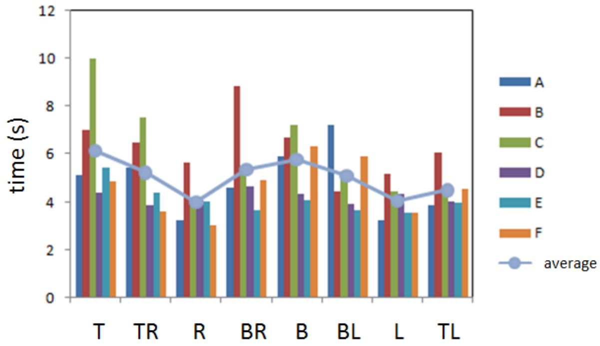


Fig. 9. Six subject A - F and average time to move depending on the face direction. T, TR, R, BR, B, BL, L and TL denote top, top right, right, bottom right, bottom, bottom left, left and top left directions, respectively.

4.3.2. Mouse click recognition by open mouth

In the previous study[1], there was a difference in the recognition rate of mouse clicks by opening the mouth depending on the distance, but in this study, mouse clicks could be recognized at any distance. The improved recognition rate is the difference in the angle of view between Kinect and the web camera and the brightness of the image. Since the angle of view of the web camera is smaller than that of Kinect, a face in the image can be obtained larger than that of Kinect and the accuracy of feature point detection is improved. In addition, the fact that the web camera was able to obtain in a brighter image compared to Kinect led to improved feature point detection accuracy.

4.3.3.Judgment accuracy of open and closed eyes

Fig.10 shows the recognition rates of open and closed eyes. Since the recognition rates exceeded 0.9 at all distances, it can be concluded that the judgment of open and closed eyes was successful. This result is better than the previous study[1]. Since the angle of view of the web camera is smaller than that of Kinect, a face in the image can be obtained larger than that of Kinect and the accuracy of feature point detection is improved. In addition, the web camera was able to obtain in a brighter image compared to Kinect led and it improved feature point detection accuracy and reduced misrecognition of open and closed eyes.

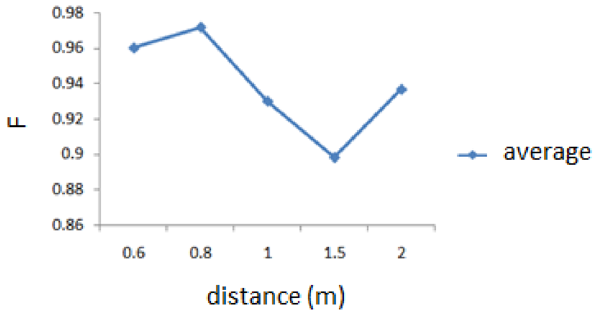


Fig. 10. F-measure of opening and closing eyes judgment.

4. Conclusion

We developed a mouse cursor control system for physically disabled people using a web camera, dlib and OpenCV. In this system, the user can move a mouse cursor depending on the face direction and perform a mouse click processing by opening the user's mouth or closing the user's eye. As the results, the mouse cursor movements took more time to move up and down compared to right and left. Moreover it is somewhat difficult for a user with glasses to operate mouse cursor compared to a user without glasses. The mouse click operation performed by opening the mouth and closing the eyes did not differ in operability depending on the distance.

The future work is to improve the mouse cursor operability of a user with glasses and add other functions. Our system has only two functions, mouse cursor movement depending on the face direction and mouse click by opening mouth and closing eyes. In order to improve practicality, it is necessary to enhance functions such as character input, double-clicking, and right-clicking. Furthermore, we would like to estimate the eye gaze and move a mouse cursor according to the eye gaze.

References

1. J. Miyachi, M. Tabuse, *"Development of Mouse System for* Physically Disabled Person by Face Movement Using Kinect", Proceedings of the 2017 International Conference on Artificial Life and Robotics, (2017), 124-127.
2. Dlib C++ Library, http:/dlib.net/
3. OpenCV, http:/opencv.org/
4. Nobuyuki Otsu, "A threshold selection method from gray-level histograms", IEEE Trans. Sys. Man. Cyber. 9 (1979) 62-66.

Authors Introduction

Dr. Masayoshi Tabuse

He received his M.S. and Ph.D. degrees from Kobe University in 1985 and 1988 respectively. From June 1992 to March 2003, he had worked in Miyazaki University. Since April 2003, he has been in Kyoto Prefectural University. His current research interests are machine learning, computer vision and natural language processing. IPSJ, IEICE and RSJ member.

Ms. Manase Mizobe

She received her B.S. degree from Kyoto Prefectural University in 2019. She works at Toray Engineering Co.,Ltd.

Dr. Yasunari Yoshitomi

He received his B.E, M.E. and PhD degrees from Kyoto University in 1980, 1982 and 1991, respectively. He works as a Professor at the Graduate School of Life and Environmental Sciences of Kyoto Prefectural University. His specialties are applied mathematics and physics, informatics environment, intelligent informatics. He is a member of IEEE, HIS, ORSJ, IPSJ, IEICE, SSJ, JMTA and IIEEJ.

Dr. Taro Asada

He received his B.S., M.S. and PhD degrees from Kyoto Prefectural University in 2002, 2004 and 2010, respectively. He works as an Associate Professor at the Graduate School of Life and Environmental Sciences of Kyoto Prefectural University. His current research interests are human interface and image processing. He is a member of HIS, IIEEJ.