Adaptable Eye tracking Technology of Obscured Face

Jian-Lung Chen[1], Yu-Ju Chao[1], Ting-Hsun Cheng[1], Yu-Hsin Lin[1]

Bigtim@itri.org.tw
[1] Industrial Technology Research Institute, Hsinchu, Taiwan 31040, R.O.C.

Keywords: Machine learning, Pupil. Partially Obscured Face.

ABSTRACT

Humans around the world are affected by special infectious pneumonia (COVID-19). There are more and more people wearing masks that are necessary for daily or workplace use. However, the sensitivity of face detection will be affected by feature obscuration, and most of them cannot be performed. Obscured face detection and gaze tracking. This paper proposes a face detection and landmark repair, and then realizes the tracking of the eye trajectory of the obscured face.

Model database with obscured face image data can also include unobscured face image data. After calibrated eye area, machine learning [1] algorithm is used for model database training to achieve eye area detection and provide real-time position coordinates. The eye information of the partial simulation model is superimposed and calculated to complete the feature point restoration, feature point detection and definition. Finally, K-means [2] is used to classify the image around the eyes to distinguish the eyeball from the white of the eye and calculate the position of the eyeball center.

The face wearing a mask will affect the sensitivity of face detection, and the person wearing a mask cannot be detected. We use a two-stage method to locate the eyeball center of the face wearing a mask.

We use the machine learning algorithm to detect the bounding box near the eyes, and we use the obscured image to train our model. Then attach the chin pattern to the place that is expected to be covered. Use a general cross-platform machine learning library [3] to locate area near the eyeball. Then use an unsupervised learning clustering algorithm to classify the image near the eyeball to analyze the eyeball area and find the center of the eyeball, to achieve the purpose of eye tracking.

1 INTRODUCTION

Generally, face detection and recognition technology is applied when the local features are occluded, and the detection and recognition rate is often poor or even failure occurs. Of course, it is not conducive to eyeball positioning and gaze tracking. To overcome this problem, this paper proposes adaptive facial features the gaze tracking technology of occlusion detection can still perform face detection and gaze tracking stably even when the facial features are occluded by 50%. In addition, an application system equipped with virtual and real interaction can provide more accurate information fusion.

2 EXPERIMENT

To perform obscured face detection and gaze tracking, a model database must be prepared. The image data is collected by using Intel RealSense D435-RGB Camera to capture close-up facial images. When discussing image resolution, higher image resolution is suitable for the accuracy of eye positioning, so the camera's highest specification 1920×1080 is used for facial image capture. In addition, the diversity and completeness of the model database helps to accurately detect the labeled area of the face, so the design of facial image capture adopts 9 sets of orientations and includes obscured and unobscured facial images, as shown in Fig. 1. Based on the design, about 5000 facial image data are initially used for machine learning of the model database.

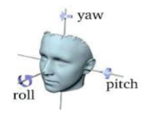
Type	T1_unobscured	T2_obscured	T3_obscured
Face rotation angle	• Pitch : -30° 、0° 、+30° • Yaw : -45° 、0° 、+45° 		
Image capture			
Image quantity	1800@1person	1800@1person	1800@1person

Fig1 Image data collected for machine learning

Before performing facial landmark restoration, we can use any unobscured face image as a basic model and define the position of its eye area, as shown in Fig. 2. When the trained model database performs user eye detection, the generated position coordinates are superimposed with the position coordinates of the model's eye area, and then the user image and the unobscured face model are superimposed into a virtual face This allows us to restore the obscured facial landmark, and then use the a general facial landmark non-obscured general cross-platform machine learning library to calculate the landmark of the real eyes in the virtual image. After obtaining the eyeball peripheral images, use the unsupervised learning clustering algorithm into K clusters, and the area belonging to the eyeball can be verified to calculate the eyeball center position and to trace the user's eyeball track. The general calculation procedure is shown in of Fig. 3.

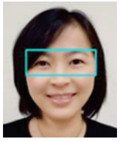


Fig. 2 Obscured face model and definition of eye region

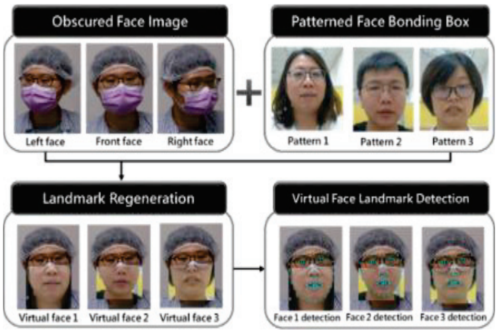


Fig. 3 Obscured landmark regeneration process

3 RESULTS

3.1 Face detection model training

After machine learning of the model database, mAP is greater than 95%, and the relationship between the number of iterations and mAP-Avg. loss error is shown in Fig. 4.

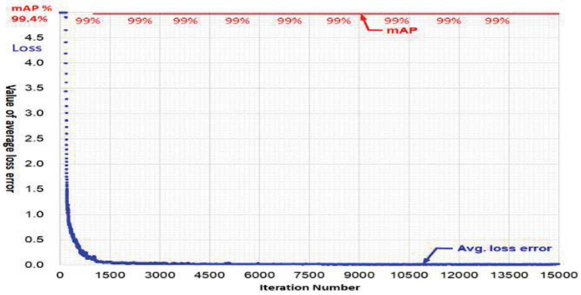


Fig. 4 Diagram mAP-AVG and loss error v.s. iteration

3.2 Eyeball center calculation

The characteristics of the eye image will be grouped according to the results of colorization. Therefore, the four color space models of RGB, HSV [4], YUV [5][6] and CIE LAB[7][8] are used to discuss the accuracy of eyeball positioning.

3.2.1 Clustering base on RGB scale level

From the superimposed virtual face image, the landmark position of the real eye can be calculated. Each eye is defined by six landmark anchor points. The hexagonal area surrounded by the six landmark anchor points does not contain skin parts. The processed image around the eyeball is shown in the green box in Fig. 5. The unsupervised learning clustering algorithm is used for the image in this area and the cluster is set to 2. The images around the eyeball are grouped to distinguish the pupil/iris and sclera. Or the circumscribes rectangle defined by the contour formed by the landmark points of the eye is shown in the blue box on the left of Fig. 5, or a relatively large rectangle can be used as shown in the red box on the right of Fig. 5. The difference between this method and the above method is the external expansion. The area covered by the rectangle will have the skin around the eyes. Therefore, the cluster value by the unsupervised learning clustering algorithm is set to 3, which means that the images in this area are grouped into three categories: the pupil/iris, the sclera, and the skin. The image is shown in Fig. 6, and the RGB brightness value is calculated according to the grouping results. The brightness value of the pupil/iris area is the lowest. After calculating the geometric center of this area is the eyeball center. It shows that the accuracy of the eyeball center point calculation is performed with different range of eye groups, and its accuracy is shown in Fig. 7.

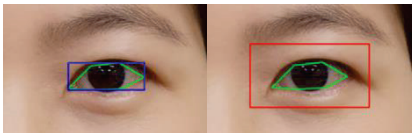


Fig. 5 Area near the eyeball

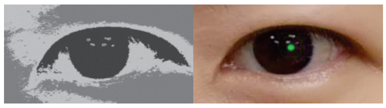


Fig. 6 Cluster and eyeball center in RGB color space

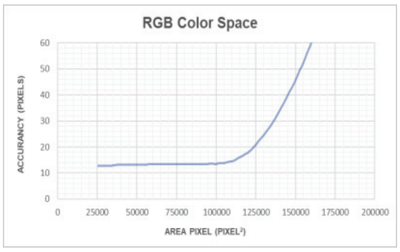


Fig. 7 Accuracy vs difference rectangle size in Y'UV color space

3.2.2 Clustering base on HSV scale level

The HSV is a color space system to represent visual colors. The HSV color space can be represented by a cone, as shown in Fig. 8 below, where the azimuthal angle represents the hue, the cone radius represents saturation, the ratio of color purity to the maximum purity of the color, and the cone height represents the brightness of the color. The color rendering is based on the conversion of the component values of each color (Red, Green, Blue) in the RGB color space, that is, the component value of each color is converted into the HSV color space, and then converted to the conical space coordinates (x, y, z), note that the x, y, z here is the coordinates we defined in the conical space, not the CIE1931 XYZ color space.

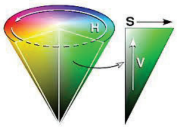


Fig. 8 The HSV color scale

Set the cone angle to 45 degrees here, and the formula for converting HSV color space into cone space coordinates is as follows

$x = s / 255 * \cos(h / 90 * \pi)$

$y = s / 255 * \sin(h / 90 * \pi)$

$z = v / 255$

For example, color red is (0, 255, 255) in the HSV color space, it is converted to (1, 0, 1) in the conical space(x, y, z). Relatively, color cyan is (90, 255, 255) in the HSV color space, it is converted to (-1, 0, 1), and the color black is (0, 0, 0) in the conical space. Therefore, after the detected eye image is finally converted to conical space, the unsupervised learning clustering algorithm is used to group the hexagon area surrounded by the six landmark anchor points with cluster is set to 3. As shown, the calculated center position of the eyeball, as shown by the green dots in Fig. 9, is also used to confirm the accuracy of the eyeball center point with different eye groups, as shown in Fig. 10.

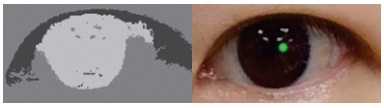


Fig. 9 Cluster and eyeball center in HSV color space

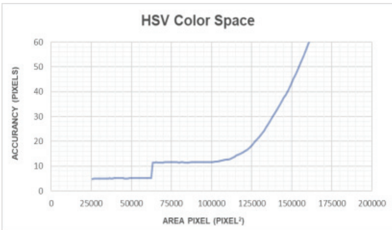


Fig. 10 Accuracy vs difference rectangle size in Y'UV color space

3.2.3 Clustering Base on Y'UV Color Scale

The YUV color space is a color encoding system, in terms of one luminance component (Y') and two chrominance components, called U (blue projection) and V (red projection) respectively. Can be converted by RGB color space, the formula based on NTSC standard is as follows:

$Y' = 0.299 \times R + 0.587 \times G + 0.114 \times B$

$U = -0.147 \times R - 0.289 \times G + 0.436 \times B$

$V = 0.615 \times R - 0.515 \times G - 0.100 \times B$

Therefore, the conversion of red in RGB color space (255, 0, 0) to Y'UV is (76.245, -37.485, 156.825). Therefore, when the RGB image is presented in the grayscale of the Y'UV color space, U and V values can present the color density in a more detailed manner. For complex colors, it is divided into groups. It is even more advantageous, as shown in Fig. 11, which shows the RGB image after Y'UV conversion. The Y'UV image is grouped by the same algorithm as above, and calculated eyeball center position is shown in the green dot in Fig. 12. Confirm the accuracy of the eyeball center point, as shown in Fig. 13.

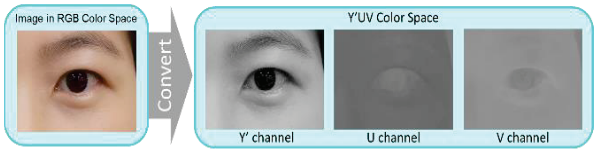


Fig. 11 Color space transform in Y'UV channel

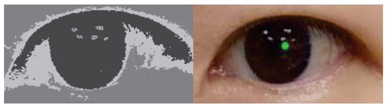


Fig. 12 Cluster and eyeball center in Y'UV color space

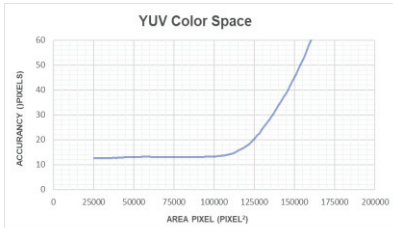


Fig. 13 Accuracy vs difference rectangle size in Y'UV color space

3.2.4 Clustering Base on CIELab Color Scale

CIELAB is the most commonly used color space system to represent visual colors. The three coordinate values L*, a*, b* are used to represent the color seen by the eye. However, when RGB color space is converted to CIELAB color space, the XYZ color space coordinate system is required as a color conversion intermediary procedure, that is, RGB color space must be converted to XYZ color space first, and then converted to CIELAB color space. The conversion formula is as follows

Convert RGB color space to XYZ color space

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} gamma(\mathrm{R}/255) \\ gamma(\mathrm{G}/255) \\ gamma(\mathrm{B}/255) \end{bmatrix}$$

$$\mathrm{gamma}(x) = \begin{cases} ((x + 0.055)/1.055)^{2.4} & if\ x > 0.04045 \\ x/12.92 & others \end{cases}$$

Convert XYZ color space to CIELAB color space

$$\boldsymbol{L^*} = 116 \times \mathrm{f}(Y/Y_n) - 16$$

$$\boldsymbol{a^*} = 500 \times [f(X/X_n) - f(Y/Y_n)]$$

$$\boldsymbol{b^*} = 200 \times [f(Y/Y_n) - f(Z/Z_n)]$$

where $\mathrm{f}(t) = \begin{cases} t^{1/3} & if\ t > (6/29)^3 \\ 1/3 \times (29/6)^2 \times t + 4/29 & others \end{cases}$

and $X_n = 95.0489,\ Y_n = 100,\ Z_n = 108.884$

The image of the same RGB eye image after CIELAB color space conversion is shown in Fig. 14. The images are grouped by the same algorithm, and the calculated eye center position is shown in the green dot in Fig. 15. The eye center point accuracy performance is performed for the different range of eye groups, as shown in Fig. 16.

Fig. 14 Color space transform in CIELAB channel

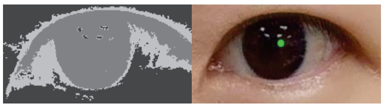


Fig. 15 Cluster and eyeball center in CIELAB color space

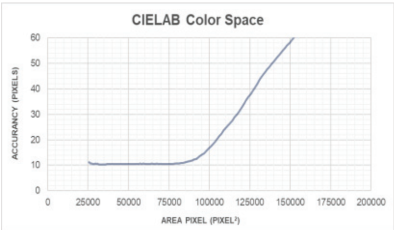


Fig. 16 Accuracy vs difference rectangle size in CIELAB color space

3.3 Eyeball center reference calculation

We refer to Mezher [9] and Ahmed [10]. Perform gray-scale conversion and Gaussian blurring with the same eye image, and finally obtain the eyeball landmark point and the center reference point, and detect the gray-scale change from dark to bright according to the preset point angle, which is defined The point is the boundary point of the pupil/iris and sclera, and then the eyeball center point of these boundary points, the result is shown in Fig. 17.

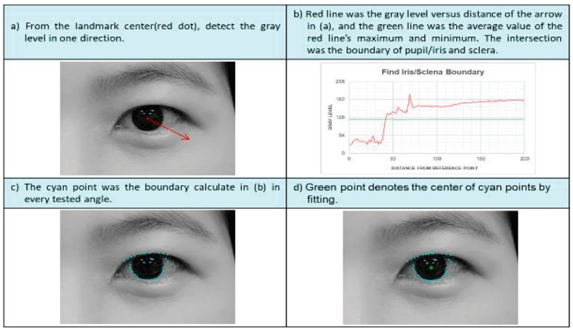


Fig. 17 Calculation of Eyeball center reference

4 DISCUSSION

Each color space model uses the unsupervised learning clustering algorithm to group the eye images according to various areas. After a series of tests, the smallest circumscribed rectangle defined by the contour formed by the landmark of the eye is accurate. After calculating the deviation from the reference value of the eye center, the error value of the RGB and YUV color space grouping for the eye center calculation is up to 12 pixels, the error value of the Lab color space grouping is 10 pixels, and the HSV The minimum error of the color space grouping is only 4 pixels, and the accuracy of the eyeball center is the best among the four color space groups.

5 CONCLUSIONS

The eye tracking technology of adaptive facial feature occlusion detection combines the machine learning algorithm to detect the eyes of the occluded face. The occluded features are repaired by image superimposition, and the general cross-platform machine learning library is used for facial landmark detection. Measure the anchor point of the eye image, and use the color space grouping technology to calculate the eyeball center point. The test verifies that the eyeball center accuracy has reached the level of 4 pixels.

REFERENCES

[1] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. Computer Science, arXiv: 1804.02767. http://arxiv.org/abs/1804.02767

[2] Jin X., Han J. (2011) K-Means Clustering. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_425

[3] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009

[4] (2008) HSV Color Space. In: Li D. (eds) Encyclopedia of Microfluidics and Nanofluidics. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-48998-8_656

[5] Podpora, Michal. (2009). YUV VS. RGB--A COMPARISON OF LOSSY COMPRESSIONS FOR HUMAN-ORIENTED MAN-MACHINE INTERFACES. Zeszyty Naukowe. Elektryka. 55-56.

[6] Szeliski R., 'Computer Vision: Algorithms and Applications', Springer, 2011.

[7] Hunter, Richard Sewall (July 1948). "Photoelectric Color-Difference Meter". JOSA. 38 (7): 661. (Proceedings of the Winter Meeting of the Optical Society of America).

[8] Hunter, Richard Sewall (December 1948). "Accuracy, Precision, and Stability of New Photo-electric Color-Difference Meter". JOSA. 38 (12): 1094. (Proceedings of the Thirty-Third Annual Meeting of the Optical Society of America).

[9] Mezher, Alhamzawi Hussein Ali. "Real Time Gaze Estimation for Medical Field using Normally Webcam with OpenCV." International journal of scientific research in science, engineering and technology 3 (2017): 136-141.

[10] Ahmed, Noha. (2020). Real-time accurate eye center localization for low-resolution grayscale images. Journal of Real-Time Image Processing. 10.1007/s11554-020-00955-2

[11] Bradski, G. (2000) The OpenCV Library. Dr. Dobb's Journal of Software Tools.