Non-intrusive Eye Gaze Estimation from a System with Two Remote Cameras

Yu Zun Neoh, Haidi Ibrahim

Abstract*: Eye gaze is the direction where a person is looking at. It is suitable to be used as a type of natural Human Computer Interface (HCI). Current researches use infrared or LED to locate the iris of the user to have better gaze estimation accuracy compared to researches that does not. Infrared and LED are intrusive to human eyes and might cause damage to the cornea and the retina of the eye. This research suggests a non-intrusive approach to locate the iris of the user. By using two remote cameras to capture the images of the user, a better accuracy gaze estimation system can be achieved. The system uses Haar cascade algorithms to detect the face and eye regions. The iris detection uses Hough Circle Transform algorithm to locate the position of the iris, which is critical for the gaze estimation calculation. To enable the system to track the eye and the iris location of the user in real time, the system uses CAMshift (Continuously Adaptive Meanshift) to track the eye and iris of the user. The parameters of the eye and iris are then collected and are used to calculate the gaze direction of the user. The left and right camera achieves 70.00% and 74.67% accuracy respectively. When two cameras are used to estimate the gaze direction, 88.67% accuracy is achieved. This shows that by using two cameras, the accuracy of gaze estimation is improved.

Index Terms*: Eye gaze, human computer interaction, iris detection, two cameras system.

I. INTRODUCTION

Human beings use their eyes to visualise things around them and to communicate with surrounding peoples. More than half of the human sensory information comes from the eye, so the eye gives many information to the user. By estimating the eye gaze, a person's presence, attention, focus, drowsiness, consciousness or other mental states can be identified [1-5]. Besides, eye gaze can also be used as a type of Human Computer Interaction (HCI) that helps human, as a user, to communicate with machine or devices. HCI that is based on eye gaze enables the user to interact with devices in a more natural way as compared to interactions through a mouse or a keyboard [6-7].

Two areas which are the industrial field and the medical field will benefit by using eye gaze as an interface for HCI. For example, in a process plant, accidents often occur. This leads to large losses which are worth up to millions in United States Dollar (USD). Analysis shows that human errors are the primary cause for more than 70.00% of accident from the past [8]. By using eye gaze interface, this will allow the operator to interact with the computer without even touching the computer. This will help in cases where operators' hands are full with other stuffs but at the same time they still need to interact with the computer. Eye gaze can also serve as a preventive measure for the operators who are controlling the machine. This is because, by monitoring the eye gaze of the operators, the performance of operators is also monitored and this can reduce the errors caused by operators [8].

The second area is in the medical field. Eye gaze based HCI will bring huge benefit to people who had been injured or handicapped, and had lost control in their hand. Using eye gaze interface, they can control the computers or machine without the need of using hands [9-11]. Besides, eye gaze is also able to revolutionize the mental health screening. This is because the information that are related to the symptoms of some mental health issues can be observed from the eye gaze [12]. Eye gaze also has been used for eye cancer treatment and in surgery procedure [13-14].

Current eye gaze researches have several weaknesses. The first weakness is that, to get an accurate location of the iris, intrusive methods such as locating infrared or LED lights near to the iris are normally being employed. Infrared and LED are considered intrusive to the human eye because when the eye is exposed to these light sources for a long period, they might cause damage to the cornea and the retina of the eye [15].

Intrusive methods normally have a better performance compared to non-intrusive methods [16]. This is because by using intrusive methods, infrared is exposed to the eye and the system will then track where the iris is looking at [9]. Non-intrusive methods used to obtain the iris location do not rely on infrared or LED, so the quality of image captured is lower as compared with by using intrusive methods. Lower quality of images captured leads to lower accuracy in gaze estimation.

Remote gaze estimation system usually has the camera attached to the screen of the system and is tracking the user's gaze remotely. While for wearable gaze estimation system, the system usually has the camera attached to the user and the camera is placed very near to the user's eye. The second weakness is that the performance of the remote gaze estimation system might not be as good as the wearable gaze estimation system. This is because for wearable gaze estimation system, with the camera put so closely to the eye, it is able to capture less noisy images [17]. Besides, head mounted camera is able to capture images that contains much information about the eye [18]. On the other hand, by using a remote camera, the eye region on the image is relatively smaller, which means there is less

Revised Manuscript Received on May 06, 2019

Yu Zun Neoh, School of Electrical & Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Seberang Perai Selatan, Penang, Malaysia.

Haidi Ibrahim, School of Electrical & Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Seberang Perai Selatan, Penang, Malaysia.

information being captured as compared to the head mounted camera. The term remote camera refers to camera which are not attached to the user.

This research aims to have a non-intrusive method to locate the iris location using remote camera. This will let the user to have a comfortable experience in having their iris located. To overcome the weakness that existing researches are facing, this research proposed to use two cameras to overcome the weaknesses discussed. By using two cameras, more data of the eye can be collected. With more data for the system to analyze, the accuracy of the system can be improved. Besides, the secondary camera will also help to act as a verification for the first camera. This manuscript is divided into four sections. Our methodology is given in Section II. Then, our experimental results are presented in Section III. Section V concludes our findings.

II. METHODOLOGY

Figure 1 shows the block diagram of our proposed eye gaze estimation system. This eye gaze estimation system is divided into 3 subsystems which are the image collecting system, the image processing system and the Graphical User Interface (GUI). The image collecting system consists of the camera and the laptop. The image collecting system is designed to capture the user's image and send to the laptop. The image is then collected to be used for processing. Two cameras are used to collect the image. By using two cameras, the captured area will be widened. With the help of a secondary camera, it also acts as a verification for the first camera, hence increase the accuracy of the system.

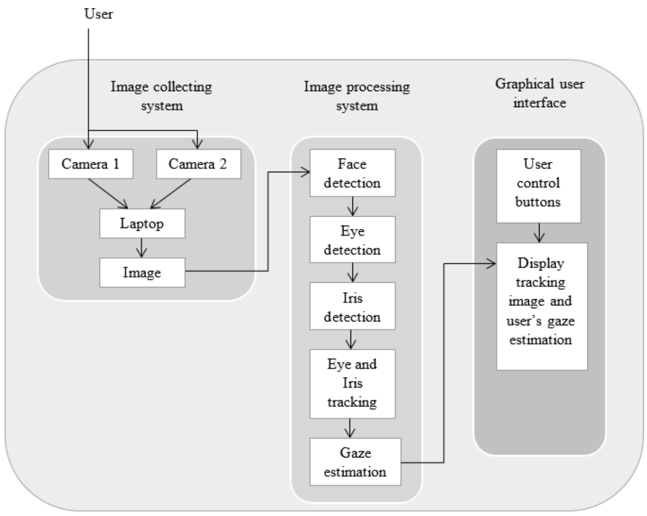


Figure 1: Block diagram of the proposed eye gaze estimation system.

The second subsystem is the image processing system. There are five stages in the image processing system which are face detection, eye detection, iris detection, eye and iris tracking and gaze estimation of the user. This system receives the image from the image collecting system. This system contains the algorithm used to track the iris and the eyes of the user and to estimate the user's gaze.

The last subsystem is the graphical user interface. The graphical user interface is designed to display the tracking information of the user's eye and iris location. The user will be able to know whether the system has successfully tracked their eye and iris location by looking at the GUI. The user's gaze estimation will also be displayed. Besides, graphical user interface also allows the user to control whether they want to start or stop the eye gaze tracking system. There is also option for user to input a captured image and the system will provide the gaze estimation of the user in the image.

Figure 2 shows the overall hardware setup. The hardware is made up of two PS3 eye cameras by Sony and one Hewlett Packard laptop. PS3 eye camera is chosen to be the camera for this system because it can capture video at frame rates of 60 Hertz at 640×480 pixels, and 120 Hertz at 320×240 pixels. With the high frame rates of this camera, it allows this system to capture the image of user smoother and faster compared to other ordinary webcams which normally have frame rates around 10 to 60 frames per second. Although the resolution of this camera is not so high, but the quality is sufficient for the image processing system to process and identify the user's face, eye and iris. The high frame rate and normal resolution of this camera makes it suitable to be used for real time tracking purposes.

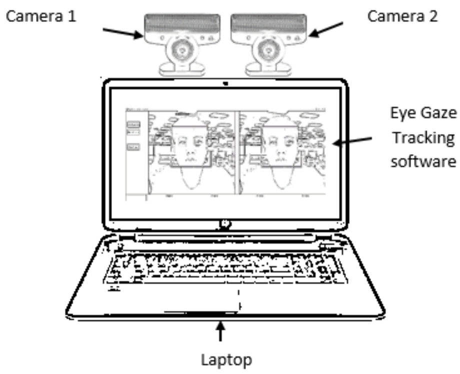


Figure 2: Overall hardware setup.

The two cameras are placed side by side on top of the screen of the laptop. This allows the camera to be in parallel with the user and it also has a better vision to capture the image of the user. This also ensures that both cameras can capture the user's both eyes when the user is looking at the screen. The laptop used for this system has 8 GB of RAM and an Intel i5 processor. This is not the highest specification, but it is sufficient to support two cameras which are needed by this system.

This system uses two cameras to locate the iris of the user. By using two cameras, the second camera will act as verification to improve the eye gaze estimation performance. If one camera loses track of the user, there is another camera that will keep track of the user. The captured area is also widened by the use of two cameras. This allows the user to have a bigger movement while the system is still able to keep track of the user. The system is designed to capture the eye and iris of the user at a distance of 40cm to 50cm. In a nutshell, the hardware designed will help to improve the accuracy of the eye gaze estimation system.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

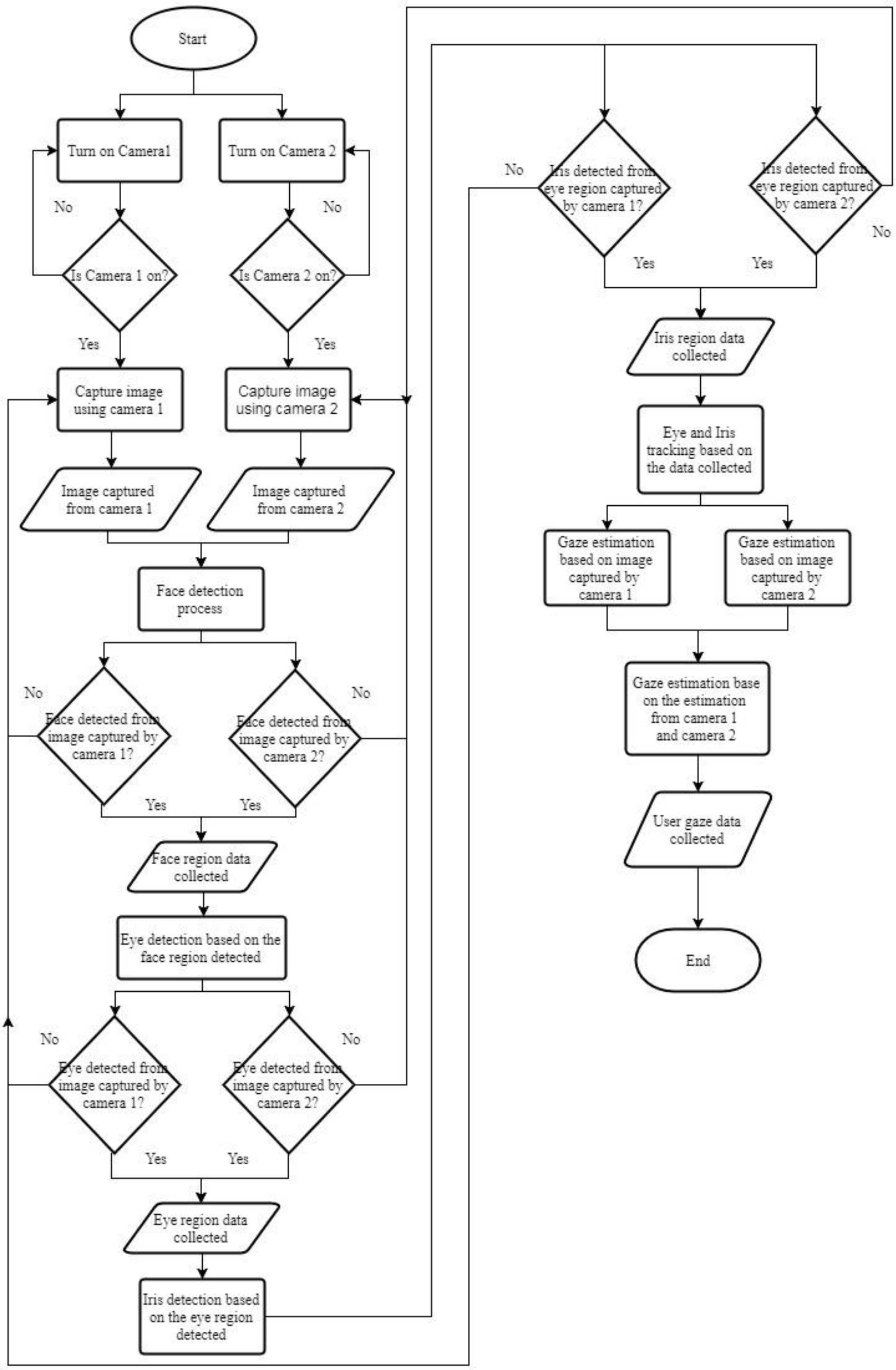


Figure 3: Process flow overview of eye gaze tracking system.

Figure 3 shows the flow chart of the eye gaze estimation system. This system uses Microsoft Visual Studio 2013 integrated development environment (IDE) to develop. This is because Visual Studio is one of the best development environments available. The programming language chosen is C#. C# is object oriented and is suitable to design applications that run on the .NET framework. The image processing libraries used are OpenCV and Emgu CV. Emgu CV is a .Net wrapper for OpenCV to allow the library to be called in C# environment. Both libraries used are open sourced and are free to access by the public.

From Figure 3, once the system starts, both cameras are turned on. Next, the system will verify that both cameras are turned on before proceed to the next stage. This is to make sure that both cameras are functioning. All the settings needed to setup the camera will also be set up here. The system will then trigger the camera to start capturing images. The image captured is flipped horizontally. This is because when capturing the user's image, the right eye will be on the left side and the left eye will be on the right side. By flipping the image horizontally, the eye in the image captured will be on the same side as the user. This will remove any confusion to the user. Once the images are captured and flipped then the image is passed to the image processing subsystem to process.

The first process is the face detection process which at this stage the system will identify whether the image contains any face. When no face is detected, then the system will capture a new image and repeat the process. If a face is detected, then the region of the face is saved to be used for the eye detection. By using the face region to find the eye, the processing time will decrease compared to search for the whole image for the eye This is because the system will need to find the eye in a larger area. When the eye is detected, the system will store the eye region and proceed to find the iris from the eye region.

If both eyes and iris location are determined, the system will proceed to save the location and use the location to track the eye and iris. The use of tracking is to speed up the processing time. Tracking allows the system to remember the last location where the eye and iris are found and the system will try to find the eye and iris near to the last location where they are previously detected in the new image. After tracking, the user's gaze is estimated. The system will determine where the user is looking at by using the location of the eye and iris. The gaze estimation and tracking information will then be displayed out in the graphical user interface.

Camera 1 and camera 2 share the same algorithm to detect the face, the eye and iris but they are two separate processes and they work independently. This is to allow two sets of data for gaze estimation. Camera 1 will come out with one set of gaze estimation of the user and camera 2 will come out with another set of data. The eye gaze tracking system will then compare both data and come out with the most accurate estimation for the direction of a user's eye gaze.

Overall, this system will keep on capturing new image at a fixed interval. This enables the system to track the eye gaze in real time mode. Besides, by using two cameras and having two separate processes for each camera will provide the system with two data; to calculate and compare, and give gaze estimation with higher accuracy.

A. Graphical User Interface

Figure 4 shows the design layout of the graphical user interface (GUI). On the left side of the GUI, there are control buttons for the user to control the settings on the GUI. There are four buttons on the GUI. The first button is the Start button which is used to start the eye gaze estimation process. There is also a stop button designed so the user can stop the system from estimating their eye gaze. There is an open image button which is designed for the purpose that the user has the option to upload an image and the system will still detect whether there are face, eye and iris and provides the gaze estimation to the user based on the image uploaded into the system. The last button is designed to let the user have the option to save the image from the result of estimation.

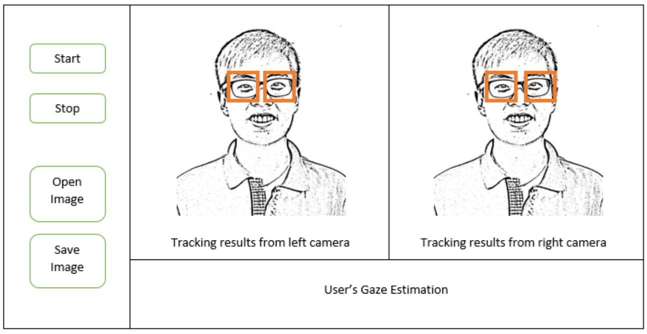


Figure 4: GUI of eye gaze estimation system.

On the right side the GUI, it shows the result of tracking on the screen. It will label the eye and the iris location of the user. Through labelling, the user will know whether the system has successfully tracked their eye. There are two results displayed here. The left side will display the estimation results from the left camera while the right side will display the estimation results from the right camera.

At the bottom part, the GUI will show the gaze estimation of the user. This GUI is designed to be very simple as the main purpose is to prove that the system can detect the eye and iris, track them and use the information to estimate the eye gaze of the user.

B. Face Detection

Figure 5 shows the process flow of face detection. First the image captured by the camera will be retrieved by the system. Next the image is converted to grayscale and the image histogram is equalized. These two steps are very important as it will speed up the processing time and increase the detection accuracy. By converting to grayscale, the system processes less information as compared to a color image so it leads to faster processing time. Image histogram equalization allows area with lower contrast to have higher contrast and it helps to improve the accuracy of detection. After the image is processed, the cascade classifier will then search for the face region in the image. If there is a face region found in the picture, then the face region location is saved to use for the next process which is to detect the eye. If there is no face region found, then the system will recapture a new image.

This system uses Viola Jones algorithm which is also known as Haar cascade algorithm to detect the face region [16,19]. To apply this algorithm, a trained classifier to detect the face is needed. This system uses an existing trained classifier which is "*haarcascade_frontalface_alt2.xml*". This classifier works best at detecting the frontal face of the user but has limitations on detecting the side faces of the user. When the user is interfacing with the computer, most of the time the user's frontal face will be in parallel with the camera so this classifier is suitable for this system.

The classifier can then be applied to the image to

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

search for possible face region. The classifier will search for the entire image using different scaling factor. This is to allow the classifier to find the objects of interest at different size. The algorithm contains several classifiers to be applied to the image to detect the face. The region of the face is then saved when a face is detected.

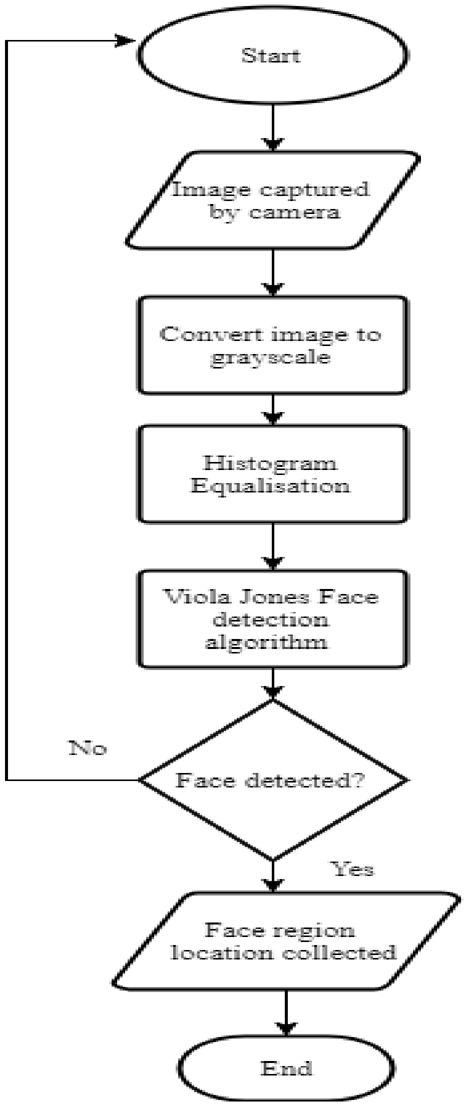


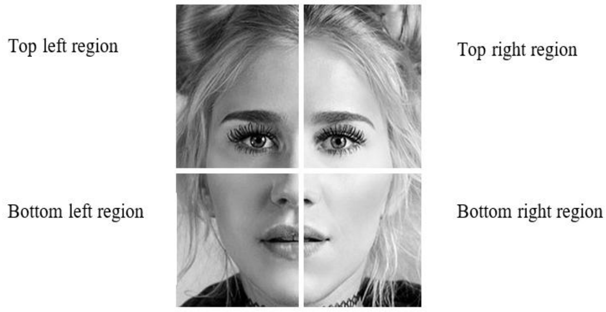
Figure 5: Face detection process flow.

C. Face Detection

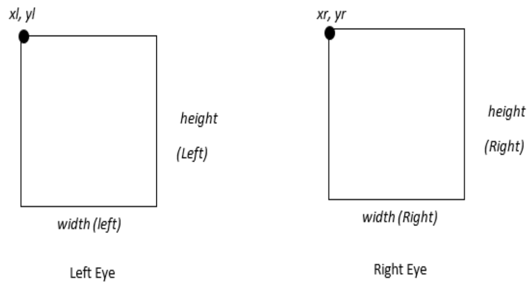
This system uses a unique approach to detect the eye, which is to have separate detection for left and right eye using Haar cascade algorithm [19]. This approach enables the system to store two separate parameters and is easier to retrieve for gaze estimation process. This system uses "*haarcascade_eye.xml*" classifier to detect the eye region.

In Figure 6(a), the figure shows that the face is separated into four regions which is the top left region, the top right region, the bottom left region and the bottom right region. The eye detection will be carried out on the top left region and the top right region where the two eyes are located in. This helps to reduce the region that the classifier needs to search for the eye. If the eye is in the region, then the parameters for both eye region are then saved separately.

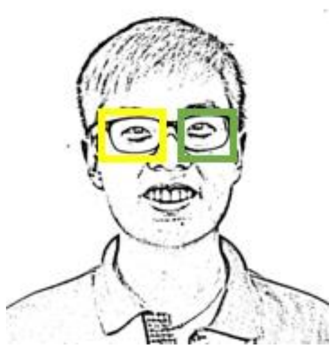
Figure 6(b) shows the parameters saved for the left eye and right eye region. The system will save the parameter separately to make it easier to retrieve for the next process. Figure 6(c) shows an example of how the system shows that the eye region is detected. The yellow color represents the left eye while the green color represents the right eye. Both eyes are located inside the highlighted rectangle which means that the eye is successfully detected. After the eye regions are detected the parameters are saved to be used for iris detection. The eye region parameters are also used during gaze estimation.



(a)



(b)



(c)

Figure 6: (a) Separating face region for eye detection. (b) Parameters for left eye region and right eye region. (c) Detected eye regions.

D. Iris Detection

The iris detection uses Hough circle transform algorithm [20] to detect the iris from the eye region. The Canny threshold and the circle accumulator are two very important thresholds. The Canny threshold is set to 170, this means the system will apply the edge detector with a threshold of 170. If the threshold is set too low, many non-edges will be detected as an edge. The circle accumulator threshold is set to 25 which means the system will accumulate 25 circles at the center detection stage before identified the object as a circle. If the threshold is set too small, then many

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

false edges will be identified as a circle.

Besides the two thresholds, the maximum radius of the circle to be detected is set at not more than the height of the eye region divided by three. Through this setting, the Hough circle transform is able to handle the variation in iris sizes. This is because the height of the eye region is based on the user, so the algorithm can handle iris with different sizes. If the iris is successfully detected, then the iris region is saved. The parameters saved are the X and Y coordinates of the center of the circle, the area of the circle and the radius of the circle.

Figure 7 shows that left and right iris are successfully detected by the system. The red circle represents the left iris while the orange circle represents the right iris.

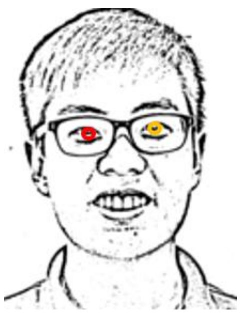


Figure 7: Detected left and right iris.

E. Eye and Iris Tracking Algorithm

This eye and iris tracking algorithm use in this system is CAMshift (continuously Adaptive Meanshift). CAMshift is an improved version of Meanshift algorithm [16,21-22]. CAMshift algorithm can continuously update the window size with the size and rotation of the tracked target.

Meanshift algorithm is an algorithm which the tracking window will set the region of interest for the tracking window. The tracking window will then analyse all the data points in the region of interest. Once the data points are identified, it will then keep the data points. The algorithm will then try to look for the same data points around the location of previous data points. Then it will highlight the data points where the algorithm calculates that they are the same as the saved data points.

For CAMshift it applies the Meanshift algorithm first, when the Meanshift is successful, it will update the size of the window. The algorithm calculates the best orientation and ellipse it. So, through this the window size will adjust to bigger size when the object is near to the camera and smaller size when the object is far from the camera. This solves the weakness of the Meanshift algorithm which the window size is fixed and cannot be changed. By applying CAMshift algorithm the eye and iris location should be able to track more consistent and will be used for the gaze estimation in the next process.

F. Gaze Estimation

The eye gaze estimation system will first estimate the user's gaze in single camera before combining the estimation from both cameras to do the final gaze estimation. The gaze estimation can estimate the direction of where the user is looking at. There are four directions which this estimation will support which is right, left, up and down.

To estimate the eye gaze, two parameters are needed. The first parameter is the location of the eye in *x* and *y* coordinates and the length and height of the eye. The second parameter is the *x* and *y* coordinates of the center of the iris. Next the estimation will be carried out as shown in Figure 8.

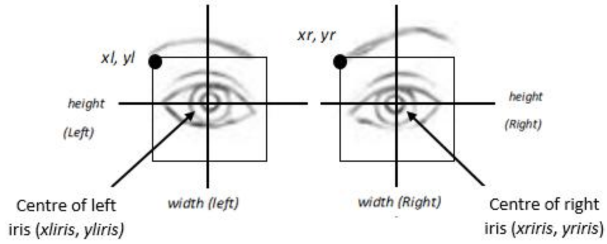


Figure 8: Parameters needed for gaze estimation

From Figure 8, it shows that the center of the iris is located at the center of the rectangle. Since the top left coordinates of both rectangles are obtained, so to get the center coordinates, the *xl* is added to the half of width and *yl* is added to the half of the height.

The gaze estimation will estimate the direction when the condition below is met:

i) Right

$$xliris > xl + \frac{width(left)}{2} + \frac{radius\ of\ left\ iris}{2} \quad (1)$$

$$xriris > xr + \frac{width(right)}{2} + \frac{radius\ of\ right\ iris}{2} \quad (2)$$

The *x* coordinate of the center of iris must be within the width of the eye.

ii) Left

$$xliris < xl + \frac{width(left)}{2} - \frac{radius\ of\ left\ iris}{2} \quad (3)$$

$$xriris < xr + \frac{width(right)}{2} - \frac{radius\ of\ right\ iris}{2} \quad (4)$$

The *x* coordinate of the center of iris must be within the width of the eye.

iii) Center (For Left and Right)

The conditions for Left and Right are not met and the *x* coordinate is within the width of the eye.

iv) Up

$$yliris < yl + \frac{height(left)}{2} - \frac{radius\ of\ left\ iris}{4} \quad (5)$$

$$yriris < yr + \frac{height(right)}{2} - \frac{radius\ of\ right\ iris}{4} \quad (6)$$

The *y* coordinate of the center of iris must be within the height of the eye.

v) Down

$$yliris > yl + \frac{height(left)}{2} + \frac{radius\ of\ left\ iris}{4} \quad (7)$$

$$yriris > yr + \frac{height(right)}{2} + \frac{radius\ of\ right\ iris}{4} \quad (8)$$

The *y* coordinate of the center of iris must be within the height of the eye.

vi) Center (For Up and Down)

The condition for Up and Down are not met and the *y* coordinate is within the height of the eye.

For the above estimation, the system will first compare whether the right and left iris provide the same estimation. If the estimations are different, the system will give priority to the left iris if the results for the gaze estimation is left or the right iris if the gaze estimation is right. The system will take the estimation from other iris if the gaze estimation from one of the irises does not meet the condition stated above.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

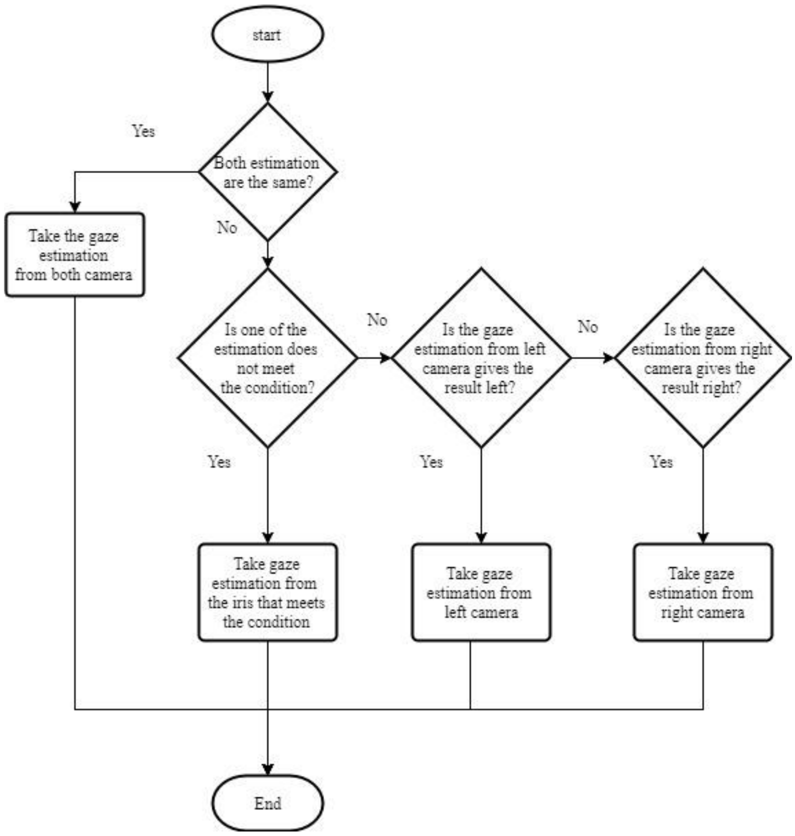


Figure 9: Process flow for two camera gaze estimation

Table 1: Results on Gaze Estimation on Left Camera and Right Camera

User's gaze direction	User 1		User 2		User 3	
	Number of frames correctly estimated by left camera	Number of frames correctly estimated by right camera	Number of frames correctly estimated by left camera	Number of frames correctly estimated by right camera	Number of frames correctly estimated by left camera	Number of frames correctly estimated by right camera
Centre	8	9	9	7	8	9
Left	8	2	8	7	8	8
Right	7	10	7	9	8	10
Up	5	7	1	6	8	9
Down	7	6	7	8	6	5

After the estimation for single camera is done, the estimation of the final gaze is calculated. By using two cameras, the captured region will be increased and more information is collected. With more information, the gaze estimation will be more accurate.

Figure 9 shows how the decision making on which gaze the system should use when combining the gaze estimation for both cameras. With left camera at the left side of the user, the camera is able to capture more clearly on the left side of the user, so the estimation will favor the gaze estimation from the left camera. With the right camera at the right side of the user then, the system will favor the estimation from the right camera if the estimation result shows that the user is looking to the right. This calculation only occurs when there is a conflict between the estimation of the both cameras. For the Up and Down estimation, it follows the calculation for the left and right. So, if the left camera gives an estimation that the user is looking to the left, then the up and down gaze estimation will favor the estimation from the left camera.

Figure 9 shows the process flow for two camera gaze estimation. From the process flow, it is shown that if there is a conflict between the gaze estimation, the camera will follow the calculation where the left camera estimates that the user is looking to the left, while the right camera gives an estimation that the user is looking to the center, then the system will choose left as the final estimation.

III. RESULTS AND DISCUSSIONS

The gaze estimation experiment is divided into two parts. The first part of the experiment is to find out the gaze estimation accuracy on the left camera and the right camera based on three different users. The results on the experiment are shown in Table 1.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

From the results obtained, in some condition the gaze estimation accuracy is very low. For user 2 the left camera only manages to estimate 1 frame correctly when user 2 gaze is at the Up position. The gaze estimation accuracy for each user's left and right camera can be calculated by using Equation (9).

$$\text{Accuracy} = (N_c/N_t)\times 100\% \quad (9)$$

where N_c is the sum of all frames correctly estimated by the corresponding camera and N_t the corresponding total frames involved in the test. Based on this formula and Table 1, the left camera gaze estimation accuracy (with $N_t = 50$) is 70.00% when tested on user 1, 64.00% when tested on user 2, and 76.00% when tested on user 3. The right camera gaze estimation accuracy (with $N_t = 50$) is 68.00% when tested on user 1, 74.00% when tested on user 2, and 82.00% when tested on user 3. The overall gaze estimation accuracy (with $N_t = 300$) for left camera is 70.00%, while for the right camera is 74.67%.

The second part of the experiment is to test the estimation accuracy of the overall gaze estimation system. The frames used are same as the frames used in the first part of the experiment which is to test the gaze estimation accuracy for the left and right camera. The results on the overall gaze estimation accuracy are shown in Table 2.

Table 2 Results on Overall Gaze Estimation using Two Cameras

User's gaze direction	User 1's number of correct frames correctly estimated by the system	User 2's number of correct frames correctly estimated by the system	User 3's number of correct frames correctly estimated by the system
Centre	8	8	10
Left	9	10	10
Right	10	10	10
Up	8	7	9
Down	8	9	7

The gaze estimation accuracy of the proposed system for each user can be calculated by using Equation (9). From this equation (with $N_t = 50$), the user 1 estimation accuracy is 86.00%, the user 2 estimation accuracy is 88.00% and the user 3 estimation accuracy is 92.00%. The overall estimation accuracy (with $N_t = 300$) when using two cameras to estimate the user's eye gaze is 88.67%.

From the results obtained, the overall gaze estimation accuracy using two cameras is 88.67%. The accuracy is higher compared to the left camera which is 70.00% accurate and the right camera which is 74.67%. This shows that by using two cameras to locate the iris of the user and estimate the user's eye gaze, the accuracy of the system is improved. The also proves that the formula designed to estimate the gaze direction of the user using two cameras has successfully increased the accuracy of the gaze estimation when compared to using one camera.

Figure 10 shows some samples on how two cameras help to increase the accuracy of the eye gaze estimation system. For User 1, the user is looking at right. The left camera gives an estimation that the user is looking at the centre while the right camera gives an estimation that the user is looking right. The final estimation is the user is looking at right. For User 2, the user is looking at the left. The left camera gives an estimation that the user is looking at the left while the right camera has lost track on one of the eyes and gives an estimation that the user is looking at the center but the final estimation is the user is looking at the left. For User 3, the user is looking at up position. The left camera gives an estimation that the user is looking at the center while the right camera gives an estimation that the user is looking up. The final estimation is the user is looking up.

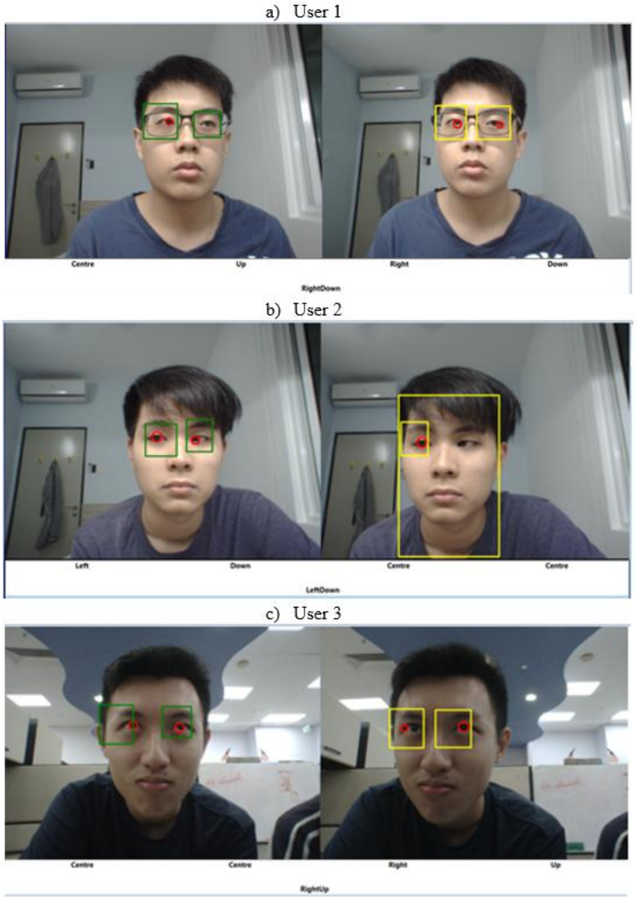


Figure 12: Some sample images from 3 users to show that the accuracy of gaze estimation is improved.

The system achieved 92.00% accuracy for user 3 when using two cameras. This is because, in the first part of the experiment, the accuracy for left and right cameras for user 3 is also the highest which are 76.00% and 82.00% respectively. The higher accuracy for left and right cameras has led to higher accuracy when using two cameras. The gaze estimation system compares the results from left and right camera before calculating the final estimation. So, with the help of higher accuracy left and right cameras for user 3, this leads to higher accuracy when using two cameras.

This experiment shows that by using two cameras to estimate the eye gaze, the accuracy of gaze estimation can be improved. The reason why by using two cameras the gaze estimation accuracy is increased, is because the system is able to collect more information on the eye and the iris of the user. With more information used to calculate the gaze estimation, the accuracy is increased.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

IV. CONCLUSION

The gaze estimation accuracy using one camera only achieved 70.00% accuracy for left camera and 74.67% for the right camera. When using two cameras the accuracy of the system achieved 88.67% accuracy. This shows that by using two cameras the accuracy of the gaze estimation increases.

ACKNOWLEDGMENT

This work was supported by the Universiti Sains Malaysia: Research University Grant 1001/PELECT/8014052.

REFERENCES

1. Panev, S. & Manolova, A., 2015. Improved multi-camera 3D Eye Tracking for human-computer interface. Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015, 1(September), pp.276–281.
2. Zheng, R., Nakano, K., Ishiko, H., Hagita, K., Kihira M. & Yokozeki, T., 2016. Eye-gaze tracking analysis of driver behavior while interacting with navigation systems in an urban area. IEEE Transactions on Human-Machine Systems, 46(4), pp. 546-556.
3. Jacob, R. J. K. 1993. Hot topics-eye-gaze computer interfaces: What you look at is what you get. Computer. 26(7), pp. 65-66.
4. Corcoran, P. M., Nanu, F., Petrescu, S, & Bigioi, P., 2012. Real-time eye gaze tracking for gaming design and consumer electronics systems. IEEE Transactions n Consumer Electronics. 58(2), pp. 347-355.
5. Hansen, D. W. & Ji, Q., 2010. In the eye of the beholder: A survey of models for eyes and gaze. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(3), pp. 478-500.
6. Liu, R., Jin, S. & Wu, X., 2006. LNCS 4282 - Single camera remote eye gaze tracking under natural head movements. Proceedings-2006 16th International Conference on Artificial Reality and Telexistence, ICAT 2006, Hangzhou, China, pp.614–623.
7. Murata, A., 2006. Eye gaze input versus mouse: Cursor control as a function of age. International Journal of Human-Computer Interaction, 21(1), pp. 1-14.
8. Kodappully, M., Srinivasan, B. & Srinivasan, R., 2016. Towards predicting human error: Eye gaze analysis for identification of cognitive steps performed by control room operators. Journal of Loss Prevention in the Process Industries, 42, pp.35–46.
9. Hiley, J.B., Redekopp, A.H. & Fazel-Rezai, R., 2006. A low cost human computer interface based on eye tracking. In 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'06, August 30, 2006 - September 3, 2006. pp. 3226–3229.
10. Nguyen, Q. X., and Jo, S., Electric wheelchair control using head pose free eye-gaze tracker. Electronics Letters. 48(13), pp. 750-752.
11. Lamti, H. A., Ben Khelifa, M. M., Gorce, Ph., & Alimi, A. M., 2013. A brain and gaze-controlled wheelchair. Computer Methods in Biomechanics and Biomedical Engineering. 16(1), pp. 128-129.
12. Itti, L., 2015. New eye-tracking techniques may revolutionize mental health screening. Neuron, 88(3), pp.442–444.
13. Elvazi, S., Bednarik, R., Leinonen, V., Fraunberg., M. v. u. z., &Jaaskelainen, J. E., 2016. Embedding an eye tracker into a surgical microscope: Requirements, design, and implementation. IEEE Sensors Journal. 16(7), pp. 2070-2078.
14. Wyder, S., Hennings, F., Pezold, S., Hrbacek, J., & Cattin., P. C., 2016. With gaze tracking toward noninvasive eye cancer treatment. IEEE Transactions on Biomedical Engineering. 63(9), pp. 1914-1924.
15. Earman, A.M., 2016. Eye Safety for Proximity Sensing Using Infrared Light-emitting Diodes, Application Note 1737,pp.1-11.
16. Santos, R. et al., 2014. Eye gaze as a human-computer interface. Procedia Technology, 17, pp.376–383.
17. Lee, H.C. et al., 2013. Remote gaze tracking system on a large display. Sensors, 13(10), pp.13439–13463.
18. Kocejko, T., Bujnowski, A. & Wtorek, J., 2009. Dual camera based eye gaze tracking system. Proceedings-2008 4th European Conference of the International Federation for Medical and Biological Engineering, pp.1459–1462.
19. Viola, P. & Jones, M.J., 2001. Robust Real-time Object Detection. International Journal of Computer Vision, (February), 57(2), pp.1–30.
20. Wang, F. et al., 2010. Face location and eye character detection based on monocular vision gaze tracking system. In Proceedings - 2010 2nd IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2010. pp. 995–999.
21. Bradski, G.R., "Real time face and object tracking as a component of a perceptual user interface," Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on, pp.214,219, 19-21 Oct 1998.
22. Huang, Y., Sang, N., Hao, Z., & Jiang, W., 2013. Eye tracking based on improved camshift algorithm. 2013 Sixth International Symposium on Computational Intelligence and Design. pp. 24-29.