CURSOR CONTROL USING EYEBALL MOVEMENT WITH RASPBERRY PI

R.Rithi[1], V.Manjuarasi[2], M.Yesodha[3], G.Renuka[4]

[1] R.Rithi Assistent Professor, ECE, PRINCE Dr.K.Vasudevan college of engineerin and technology, Tamilnadu,India
[2] V.Manjuarasi Assistant Professor, ECE, Prince Dr.K.Vasudevan college of engineering and technology, Tamilnadu, India
[3] M.Yesodha Assistant Professor, ECE, Prince Dr.K.Vasudevan college of engineering and technology, Tamilnadu, India
[4] G.Renuka Assistant Professor,ECE, Prince Dr.K.Vasudevan college of engineering and technology, Tamilnadu, India

ABSTRACT

Some people cannot operate computers because of some illness. The idea of eye controls is of great use to not only for the future of natural input but more importantly to the handicapped and disabled. Moreover implementing a controlling system in it enables them to operate computer without the help of another person. This gadget is most useful for person who can operate cursor by movement of eye. In this project Camera is used for capturing the image of eye movement. It first detect pupil centre position . Then the different variation on pupil position get different movement of cursor. The Implementation process for Pupil detection is done using Raspberry pi and on the terminal of Raspier image installed on raspberry pi. The Raspberry Pi is a credit card sized single computer or SoC uses ARM1176JZF-S core. SoC, or System on a Chip, is a method of placing all necessary electronics for running a computer on a single chip. Raspberry Pi needs an Operating system to start up. In the aim of cost reduction, the Raspberry Pi omits any on-board non-volatile memory used to store the boot loaders, Linux Kernels and file systems as seen in more traditional embedded systems.

Keyword: - Gadget , SoC, pupil, and Raspbian image

1. INTRODUCTION

Nowadays personal computer systems are carrying a huge part in our everyday lives as they are used in areas such as work, education and enjoyment. What all these applications have in common is that the use of personal computers is mostly based on the input method via keyboard and mouse. While this is not a problem for a healthy individual, this may be an insurmountable bound for people with limited freedom of movement of their limbs. In these cases it would be preferable to use input methods which are based on more abilities of the region such as eye movements. To enable such substitute input methods a system was made which follows a low-price approach to control a mouse cursor on a computer system. The eye tracker is based on images recorded by a mutated webcam to acquire the eye movements. These eye movements are then graphed to a computer screen to position a mouse cursor accordingly. The movement of mouse by automatically adjusting the position of eyesight. Camera is used to capture the image of eye movement. In general, any digital image processing algorithm consists of three stages: input, processor and output. In the input stage image is captured by a camera. It sent to a particular system to focus on a pixel of image that's gives, its output as a processed image. Embedded system is combination of hardware and software.An embedded system can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke Python is a high-level language. This means that Python code is written in largely recognizable English, providing the Pi with commands in a manner that is quick to learn and easy to follow. This is in marked contrast to low-level languages, like assembler, which are closer to how the computer "thinks" but almost impossible for a human to follow without experience.

2. RELATED WORK

There are two components to the human visual line-of-sight: pose of human head and the orientation of the eye within their sockets. Investigated these two aspects but will concentrate on the eye gaze estimation in this concpt. The present of novel approach called the "one-circle" algorithm for measuring the eye gaze using a monocular image that zooms in on only one eye of a person. Observing that the iris contour is a circle, Estimate the normal direction of this iris circle, considered as the eye gaze, from its elliptical image. From basic projective geometry, an ellipse can be back-projected into space onto two circles of different orientations. However, by using a geometric constraint, namely, that the distance between the eyeball's center and the two eye corners should be equal to each other, the correct solution can be disambiguated. This allows us to obtain a higher resolution image of the iris with a zoom-in camera, thereby achieving higher accuracies in the estimation. A general approach that combines head pose determination with eye gaze estimation is also proposed. The searching of the eye gaze is guided by the head pose information. The robustness of our gaze determination approach was verified statistically by the extensive experiments on synthetic and real image data. The two key contributions in this concept are that show the possibility of finding the unique eye gaze direction from a single image of one eye and that one can obtain better accuracy as a consequence of this.

The first technique is proposed to estimate the 3-D eye gaze directly. In this technique, the cornea of the eyeball is modelled as a convex mirror. Via the properties of convex mirror, a simple method is proposed to estimate the 3-D optic axis of the eye. The visual axis, which is the true 3-D gaze direction of the user, can be determined subsequently after knowing the angle deviation between the visual axis and optic axis by a simple calibration procedure. Therefore, the gaze point on an object in the scene can be obtained by simply intersecting the estimated 3-D gaze direction with the object. In addition, a dynamic computational head compensation model is developed to automatically update the gaze mapping function whenever the head moves. Hence, the eye gaze can be estimated under natural head movement. Furthermore, it minimizes the calibration procedure to only one time for a new individual. The advantage of the proposed techniques over the current state of the art eye gaze trackers is that it can estimate the eye gaze of the user accurately under natural head movement, without need to perform the gaze calibration every time before using it. Our proposed methods will improve the usability of the eye gaze tracking technology, and believe that it represents an important step for the eye tracker to be accepted as a natural computer input device.

In general, the visible image-based eye-gaze tracking system is heavily dependent on the accuracy of the iris center (IC) localization. In this paper, we propose a novel IC localization method based on the fact that the elliptical shape (ES) of the iris varies according to the rotation of the eyeball. We use the spherical model of the human eyeball and estimate the radius of the iris from the frontal and uprightview image of the eye. By projecting the eyeball rotated in pitch and yaw onto the 2-D plane, a certain number of the ESs of the iris and their corresponding IC locations are generated and registered as a database (DB). Finally, the location of IC is detected by matching the ES of the iris of the input eye image with the ES candidates in the DB. Moreover, combined with facial landmark points-based image rectification, the proposed IC localization method can successfully operate under natural head movement. Experimental results in terms of the IC localization and gaze tracking show that the proposed method achieves superior performance compared with conventional ones.

. Students' eye movements during debugging were recorded by an eye tracker to investigate whether and how high and low performance students act differently during debugging. Thirty- eight computer science undergraduates were asked to debug two C programs. The path of students' gaze while following program codes was subjected to sequential analysis to reveal significant sequences of areas examined. These significant gaze path sequences were then compared to those of students with different debugging performances. The results show that, when debugging, high-performance students traced programs in a more logical manner, whereas low-performance students tended to stick to a line-by-line sequence and were unable to quickly derive the program's higher-level logic. Low-performance students also often jumped directly to certain suspected statements to find bugs, without following the program's logic. They also often needed to trace back to prior statements to recall information, and spent more time on manual computation. Based on the research results, adaptive instructional strategies and materials can be developed for students of different performance levels, to improve associated cognitive activities during debugging, which can foster learning during debugging and programming.

Real-time driver distraction detection is the core to many distraction countermeasures and fundamental for constructing a driver-centered driver assistance system. While data driven methods demonstrate promising detection performance, a particular challenge is how to reduce the considerable cost for collecting labeled data. This paper explored semi-supervised methods for driver distraction detection in real driving conditions to alleviate the cost of labelling training data. Laplacian support vector machine and semi-supervised extreme learning machine were evaluated using eye and head movements to classify two driver states: attentive and cognitively distracted. With the additional unlabeled data, the semi-supervised learning methods improved the detection performance (*G*-mean) by 0.0245, on average, over all subjects, as compared with the traditional supervised methods. As unlabeled training data can be collected from drivers' naturalistic driving records with little extra resource, semi-supervised methods, which utilize both labeled and unlabeled data, can enhance the efficiency of model development in terms of time and cost.

3. PROPOSED METHODOLOGY

The figure.3.4 represents the proposed system, it uses Raspberry Pi board of version 3, the whole system is controlled by Arm11 processor . This board is connected with Monitor, IR Sensor and SD card. These components are connected by USB adaptors. Raspberry pi is the key element in processing module which keeps on monitors eye movement by interfacing with USB sensor. Sensor captures the motion of change in the eye movement . IR Sensor will be interface with raspberry pi. Raspberry pi will be using SD card, to install raspian Os and programming codes.

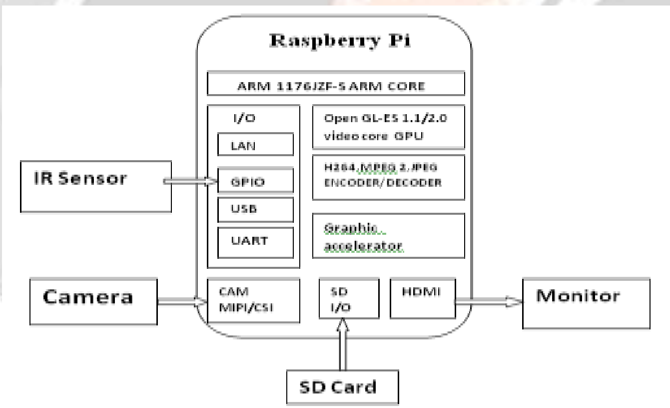

Fig 3.4 Proposed Block Diagram

3.1 IR SENSOR:

The IR Sensor used here is a movement sensor. This IR sensor is connected to the General Purpose I/O port of the raspian board, it detects the eye pupil movement and hence makes the camera to start capturing the images. Sensor can cover the range upto 5cm.

Fig 3.2 IR sensor **Fig 3.3** WebCamera

3.2 CAMERA

Once when the sensor detects the movement of eye pupil the camera starts to capture the images and send it into the raspbian board through an USB cable. The camera used here ia an USB web camera of affordable cost. Then the transferred image is processed and monitored out.

3.3 SD CARD

The function of the SD card is to store the Raspdian Jessie operating system module and it stores the program in it. The Raspberry Pi board is activated using Python programming language. The SD card capacity is upto 8GB.

3.4 MONITOR

The monitor input is got from the HDMI port of the raspberry board. HDMI is a High Definition Multimedia Interface port which is used to monitor the uncompressed video data. The HDMI converts digital image signal into analog signal and gives it to the monitor.

3.5 SYSTEM APPROACH

Camera is used to capture the image of eye movement. The pupil centre position of eye is first detected and then the different variation on pupil position is identified to get different movement of cursor. The Implemented process for Pupil detection is performed using Raspberry pi. The pupil reference has the coordinates of (x, y). Raspberry pi will be combine with USB Camera. Raspberry pi will be use SD card, then the install raspbian OS and open cv on raspberry pi. First image will be capture by USB Camera. Focus on eye in image and detect the center position of pupil by opencv code. The figure 3.5 represents this flow of system approach.

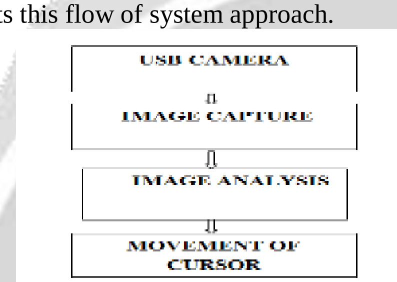

Fig 3.5 Flow Of System Approach

3.5 IMAGE PROCESSING CONCEPTS

The Image processing technologies used in this concept are:

- Haar cascading process.
- Morphological process
- Finding contours.
- Edge detection.

3.5.1 HAAR Cascading Function

One of the contributions of Viola and Jones was to use summed area tables, which they called integral images. Integral images can be defined as two-dimensional lookup tables in the form of a matrix with the same size of the original image. Each element of the integral image contains the sum of all pixels located on the up-left region of the original image (in relation to the element's position). This allows to computer sum of rectangular areas in the image, at any position or scale, using only four lookups:

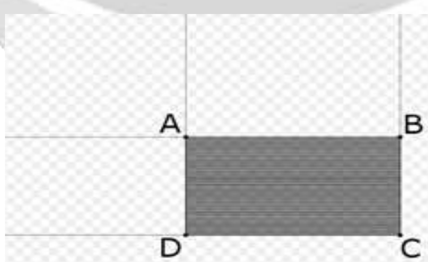

Fig 3.5.1 HAAR Pixel

3.5.2 MORPHOLOGICAL PROCESSING

The word Morphology commonly denotes a branch of biology that deals with the form and structure of animals and plants. Morphological operations are intended to affect the shape of the object. Here in this project the shape of the eye pupil is denoted using morphological analysis.*Morphology* is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbours. By choosing the size and shape of the neighbourhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbours in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion. This table lists the rules for both dilation and erosion.The following figure 3.5.2 illustrates the dilation of a binary image. Note how the structuring element defines the neighbourhood of the pixel of interest, which is circled. The dilation function applies the appropriate rule to the pixels in the neighbourhood and assigns a value to the corresponding pixel in the output image. In the figure, the morphological dilation function sets the value of the output pixel to `1` because one of the elements in the neighbourhood defined by the structuring element is on.

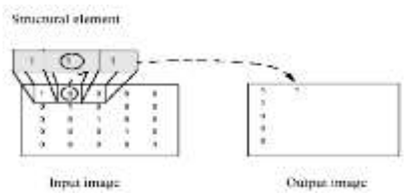

Fig 3.5.2 Morphological Dilation of a Binary Image

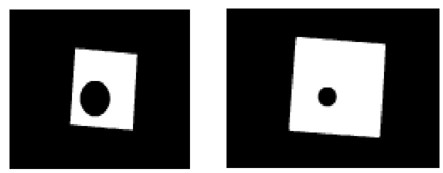

Fig 3.5.3 Morphological Erosion of a Grayscale Image

3.5.3 FINDING CONTOURS

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. For better accuracy, use binary images. So before finding contours, apply threshold or canny edge detection . Find Contours function modifies the source image. So if you want source image even after finding contours, already store it to some other variables.In OpenCV, finding contours is like finding white object from black background. So remember, object to be found should be white and background should be black.

3.5.4 EDGE DETECTION

An edge is a set of connected pixels that lie on the boundary between two regions. Edge detection is the common approach used for detecting meaningful transitions i.e, discontinuities in the grey level of an image.In this project the edges of the pupil is detected in order to note the direction in which the eyeball is rotating. There aremany ways to perform edge detection. However, the most may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image to find edges.

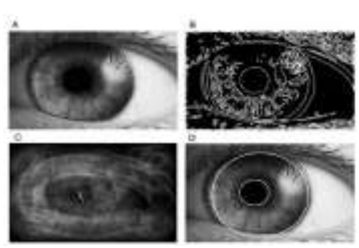

Fig 3.5.4 Various Edge Detection Filters

This first figure shows the edges of an image detected using the gradient method (Roberts, Prewitt, Sobel) and the Laplacian method (Marrs-Hildreth)Notice that the facial features (eyes, nose, mouth) have very sharp edges. These also happen to be the best reference points for morphing between two images. Notice also that the Marr-Hildreth not only has a lot more noise than the other methods, the low-pass filtering it uses distorts the actual position of the facial features. Due to the nature of the Sobel and Prewitt filters we can select out only vertical and horizontal edges of the image as shown below. This is very useful since we do not want to morph a vertical edge in the initial image to a horizontal edge in the final image.

4. RESULT AND DISCUSSION

4.1 INITIATION OF CAMERA

Initially for running camera, python coding is entered in the terminal file which is shown in figure 4.1

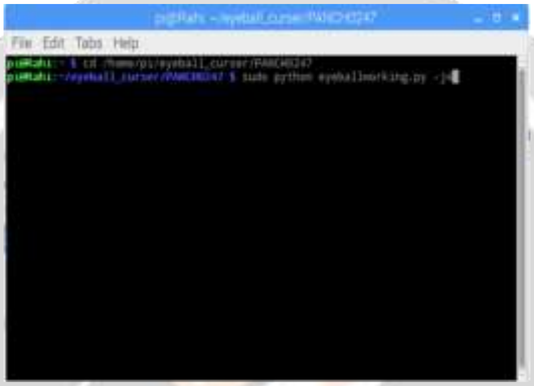

Fig 4.1 Python coded lines to initiate Camera

4.2 DETECTION OF EYE

After initiating the OS the operation starts, at first IR sensor is used to find the presence of individual in front of system then camera will be ON, where image will be capture by USB Camera. Focus on eye in image shown in figure 4.2 and detect the center position of pupil by Open CV code. Take the exact position value of pupil as reference, and then the next the different value of X, Y coordinates will be set for accurate command.

Fig 4.2 Detection of eye

Fig 4.3 Grayscale image of eye

4.3 GRAY SCALE CONVERSION OF IMAGE

Then the focused eye image is converted into grey scale image shown in figure 4.3 by image processing concept in binary form of matrix. In Processing, first we convert the color image into grey, because it is easy to process the grey image in single color instead of three colors. Gray images requires less time in processing.

4.4 CURSOR OPERATIONS

The moving position of the cursor takes the initial position as the base. As the pupil move to one of the direction, the coordinate of the mouse pointer on screen change according to the action of the pupil and the initial stage of cursor control by opencv is given in figure 4.4. Selecting the right feature plays a critical role in tracking. We are focused to track the black color in the real time video in order to track the motion of pupil.

Fig 4.4 Initial stage of cursor control

Fig 4.5 Right Move

After fixing the threshold value from 0 t0 5 then the cursor movement is based on the maximum and minimum value from that. If the pupil of eye moves to direction above the value of 2 from threshold the cursor moves right or left as per the eyeball rotating directions. This fig 4.5 shows the right side movement of the cursor with respect to the eyeball movement. Where else if the value is above 2 from threshold then the cursor moves left by the pupil direction is shown in figure 4.6.

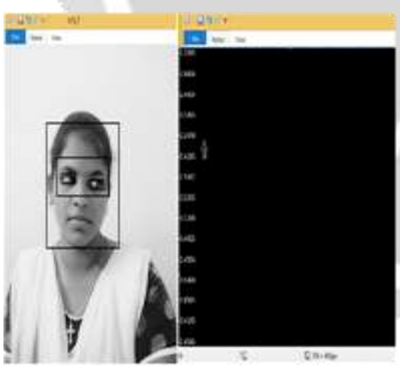

Fig 4.6 Left Move

Fig 4.7 Selection of file

Then while blinking the eye the selection of file operation take place by cursor is shown in figure 4.7.

5.CONCLUSION AND FUTURE REFERENCE

The cursor control using eye ball movement. Without using the hands we can operate the computers. This technology is contracted to replace the conventional computer screen pointing device for the use of disabled persons. The movement of cursor automatically moves by adjusting the position where of eyesight. It is mostly used for the disabled and paralysed people. Without the help of other person they can use the computers. This work can be extended to implement efficient movements to perform the click events and also cover the total mouse function of the system and to cover the total human-computer interface system using eye blink. Technology also extended to the eyeball movement and eye blinking to get the efficient and accurate movement.

In future, many people who are unable to operate a standard computer mouse or keyboard because of disabilities of their hands or arms, can get possible alternative in multimodal system, which allows controlling a computer without using standard mouse and keyboard. Using head movements to control the cursor across the computer screen and by using the speech for giving the control commands. Automatic speech recognition and head tracking in joint multimodal action are combined to operate the system.

6.REFERENCES

[1].Brooks,R.E.(1997) “Towards a theory of the cognitive processes in computer programming,” Int. J. Man-Mach. Studies, vol. 9, pp. 737–751.

[2].Cheng- Chih Wu, Ting-Yun Hou(2015)”Tracking Students’ Cognitive Processes During Program Debugging”—An Eye-Movement Approach, IEEE.

[3].Ehrlich,K. and Soloway,E.(1983) “Cognitive strategies and looping constructs: An empirical study,” Commun. ACM, vol. 26, no. 11, pp. 853–860.

[4].Eric Sung and Jian-Gang Wang (2002)“Study on Eye Gaze Estimation”, IEEE, VOL. 32, NO. 3, JUNE .

[5].Murphy,L. (2008)“Debugging: The good, the bad, and the quirky—A qualitative analysis of novices' strategies,” SIGCSE Bull., vol. 40, no. 1, pp. 163–167.

[6].QiangJi and Zhiwei Zhu (2007)”Novel Eye Gaze Tracking Techniques Under Natural Head Movement”, Senior Member, IEEE, VOL. 54, NO. 12, DECEMBER .

[7].Rajlich,V. and Xu,S.(2004) “Cognitive process during program debugging,” in Proc. 3rd IEEE ICCI, pp. 176–182.

[8].Renumol,V.(2009) “Classification of cognitive difficulties of students to learn computer programming,” Indian Inst. Technol. India.

[9].Seung-Jin Baek and Young-Hyun Kim(2013)”Eyeball Model-based Iris Center Localization for Visible Image-based Eye-Gaze Tracking Systems”,IEEE.

[10].Tianchi Liu, Yan Yang.Driver (2015)”Distraction Detection Using Semi-Supervised Machine Learning”, IEEE.