Eye Based Tracking and Control System

Jishnu M Thampan, Femitha Mohammed, Tijin M S, Pratibha S Prabhu, Rince K. M.

Abstract: ***An eye based tracking and control system is primarily used to track the human eye movements and in turn control appliances as well as a replacement for mouse. Eye movements could be tracked by tracking the position of the pupil. Real time video processing is carried out with the help of a camera which samples the images constantly and a processor. The images taken by the camera are sent to a single board computer / PC where image processing is done to identify the location of the pupil. The necessary calibration is then carried out by which cursor tracking and appliance control could be made possible. For appliance control a separate unit is fed with the control signals which select the appliance to be controlled. In order to achieve cursor, control the control signals are fed to a computer and proper calibration would help to achieve the desired output results.***

I. INTRODUCTION

The idea of object control based on eye movements is of great use particularly to the physically challenged and paralyzed people communicating with the surrounding environment. The eye gaze provides a very efficient way of pointing. Eye tracking technology involves making use of our gaze in interaction with embedded systems. Eye tracking is not heavily used in mainstream products but is beginning to pick up as inputs to electronics become more and more natural. As mentioned, with eyesight being their guide, the disabled would save energy and could use their hands and arms for other activities and could easily interact with computer-based devices and thus be at par with others.

Eye based tracking has a large number of possible applications that benefit society. There are many obvious benefits, such as aiding the disabled or a new form of human computer interaction. Eye tracking can be used to study the user's visual behaviour when surfing the internet. The applications of this would affect advertising, website design, improving usability in computer systems and optimizing web browsers and websites for maximum user experience. By using this technology for measuring online behaviour, a possible way could be found to use it to sell more highly targeted advertising. A video could be loaded by gazing it over a long period of time. Marketers have used vision heat mapping for years to determine what bores and interests' consumers. It is thus possible for an ad to change dynamically based on where on a page glanced. This technology could be used to automatically adjust a text page on the Web to display pop-up translations and meanings of foreign or difficult words, recognize skimming behaviour and adjust the page to make keywords more prominent and common words fade out,

Revised Version Manuscript Received on April 26, 2017.

Jishnu M. Thampan, Department of Electronics Engineering, Model Engineering College, Kochi (Kerala) - 682021, India.

Femitha MohammeD, Department of Electronics Engineering, Model Engineering College, Kochi (Kerala) - 682021, India.

Tijin M. S, Department of Electronics Engineering, Model Engineering College, Kochi (Kerala) - 682021, India.

Pratibha S. Prabhu, Department of Electronics Engineering, Model Engineering College, Kochi (Kerala) - 682021, India.

Rince K. M, Department of Electronics Engineering, Model Engineering College, Kochi (Kerala) - 682021, India.

And even measure what types of things people tend to skip over. Another field of eye tracking is to estimate the behaviour of the eyes while reading a book and from that information on how cognitive and lexical processing affects reading and learning was inferred. This could lead to the development of cognitive development theories that aid in teaching and learning through reading. Gaming environment could be improved with the help of eye tracking as with the help of eye movements a greater realistic and robust performance could be achieved. Also, controlling of robots could be made simple with the help of eye movements and this would increase the efficiency of robotic systems. The overview of the current system is demonstrated in figure 1.

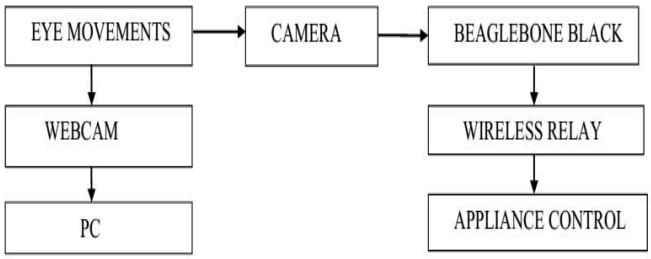

Fig. 1. General Block Diagram

II. EYE TRACKING

A. Haar Cascade Based Algorithm

An approach used to detect objects in general, applicable to human faces as well was presented by Viola-Jones. This method proved to detect objects extremely rapidly and is comparable to the best real time face detection systems. Viola and Jones (2004) presented in their research a new image representation called Integral Image which allows fast calculation of image features to be used by their detection algorithm. The second step is an algorithm based on Ada Boost which is trained against the relevant object class to select a minimal set of features to represent the object. The third step is introducing series of classifiers which have an increasing complexity. Viola and Jones used features extracted from the training set and Ad boost algorithm to select the best feature set and constructing the final classifier which comprises few stages. Each stage consists of few simple weak classifiers that work together to form a stronger Vclassifier filtering out the majority of false detections at early stages and producing an adequate final face detector.

Specific Haar-like features of a human face are looked upon by the face detection algorithm. Upon a successful match, the algorithm allows the face candidate to pass to the next stage of detection. A face candidate is a rectangular section of the original image called a sub-window. Generally, these sub-windows have a fixed size (typically 24×24 pixels). This sub-window is often scaled in order to obtain a variety of different size faces. The algorithm scans the entire image with this window and denotes each respective section a face candidate.

The algorithm uses an integral image in order to process Haar features of a face candidate in constant time.

It uses a cascade of stages which is used to eliminate non-face candidates quickly. Each stage consists of many different Haar features. Each feature is classified by a Haar-feature classifier. The output generated by the Haar feature classifiers can then be provided to the stage comparator. The stage comparator sums the outputs of the Haar feature classifiers and compares this value with a stage threshold to determine if the stage should be passed. The face candidate is concluded to be a face once all the stages are successfully passed through.

It uses a decision tree composed of a cascade of Haar-like feature classifiers to scan through the image and decide if faces are present. A Haar-like feature is generated by calculating the differences in pixel intensities in a certain region of interest. Certain pixel intensity differences can be more or less common to typical facial structures, and thereby be used to decide if the region of interest contain a face or not. For example, a typical face viewed from the front will have two darker regions for the eyes, separated by some lighter space. A single classifier decides if one of these common characteristics is met. Since many of these Haar-like features would need to be present to decide with a high degree of confidence that a face is present, the individual classifiers are cascaded into a large decision tree that is applied to every region within an image. The classifier cascade is created by training with both positive images (those with faces) and negative images (those without faces). In this training operation, the individual Haar-like features are determined and a cascaded decision tree formed, based on the similar pixel intensity difference aspects of known faces and in the absence of faces. In this way, the cascade can be customized to certain kinds of faces, faces at a certain angle, etc. This training process is very time consuming, so, fortunately a cascade trained to general frontal face detection is provided with OpenCV. The cascade is stored as a .xml file and passed into the facial detection program.

B. Face Detection

Face detection and localization is the task of checking whether the given input image contains any human face, and if so, returning the location of the human face in the image. It detects facial features and ignores anything else, such as buildings, trees and bodies Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Face detection is difficult mainly due to a large component of non-rigidity and textural differences among faces. The great challenge for the face detection problem is the large number of factors that govern the problem space. The long list of these factors includes the pose, orientation, facial expressions, facial sizes found in the image, luminance conditions, occlusion, structural components, gender, ethnicity of the subject, the scene and complexity of image's background.

Face detection is necessary to minimize the amount of error involved in eye detection. This is achieved as the eye detection part is made such that it works only after successful face detection

The face detection is the first step in the implementation of the system. It was decided to use Haar detection for this step. The first necessity is to load the classifiers. This can be done using the below code. The classifier used for the face detection was haarcascade_frontalface_alt2.xml. Once the cascade is loaded we can use it in the method for Haar Detection but first the image needs some minimal pre-processing. The image is grey scaled to make the Haar detection more robust. Since a grey image is more useful for the rest of the program as well, the grey image becomes the basis for the entire process.

The Haar detection method takes in an image, a Haar classifier cascade, the storage space that the answer should be stored in and then several other optional parameters. This function then returns a sequence of faces detected, from which the x and y coordinates, and the width and height of the face box can be extracted. Using these extracted values, we can determine the area that contains the face. A rectangle is then drawn to indicate the detected face.

C. Eye Detection

The eye detection is done separately for each eye Haar classifier. The method follows the same basic steps as the facial detection: use the Haar detection method, set the ROI, copy only the ROI to a new image and return that image. The next step in the process is pupil detection. This is by far the most difficult part of the process.The exact method of finding the pupil within the eye differs between algorithms but there are some consistencies. A particular technique is to detect circles in the image but this requires pre-processing of the image first. The pre-processing begins by equalizing the histogram of the image. This distributes the intensities of the lower contrast areas better and widens the differences between the light and dark areas.

The light and dark areas are now more defined. To reduce the noise the image was brightened as well. Applying a binary threshold then further accentuates these differences reducing the noise the next step of edge detection. To further reduce the noise a smoothing filter is applied. This reduces the overall noise and allows patterns, like the circle of the Iris, to be more readily noticed. After the pre-processing is done the pupil detection can be done. The function searches for circles in the image which is a form of ellipse fitting. To make the function only detect the best circle in the image we set the parameter called min_distance to the size of the image. This parameter represents the minimum distance allowed between two detected circles. The centre of the pupil can be approximated as the centre of this discovered circle.

D. System Overview

The block diagram of the system is shown below:

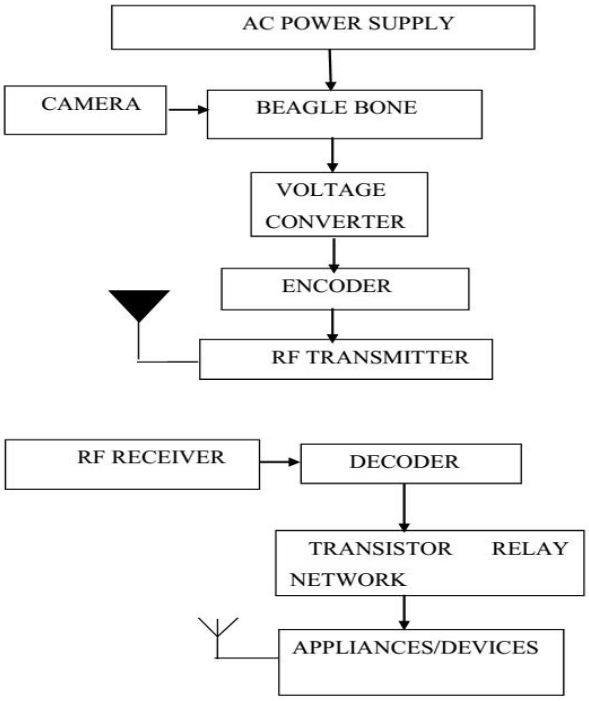

Fig. 2. System Block Diagram

As seen the whole system consists of the following units:

The image capturing unit comprises of a camera. For cursor control the webcam present in the computer serves the purpose. For appliance control, the Logitech C100 webcam is used because it connects to the host with USB and is supported by the Linux uvcvideo driver, which is included in the Angstrom Linux build.

This appliance control unit consists of several appliances which are interfaced to relays which are in turn interfaced to the GPIO pins of Beagle Bone. We use a wireless relay module to interconnect in order to convert the voltage level to 230V as the voltage from the GPIO pin would be 5V DC. Depending on the threshold values set the appliance selected would be triggered on or off with the help of on/off functions performed by the GPIO pins.

The cursor control section unit comprises of a Laptop/PC and webcam. The images to be processed are received from the web camera which is built in with the laptop. Real time video processing is done with the help of OpenCV libraries. The position of the pupil is calculated based on its coordinates and the calibrated values.

III. HARDWARE DETAILS

The primary hardware includes a Beagle Bone, wireless relay module and USB connectors and a camera. Beagle Bone is a single boar computer for real time image processing. The wireless relay module is used for controlling the appliances.

A. Beagle Bone Black

The Beagle Bone has been chosen based on its performance details which are of a high standard for an embedded board. Also, its easy compatibility with USB devices, Wi-Fi and of course webcams, make it a good choice. The image processing capabilities are enhanced by the TMS320C64X DSP and ARM Cortex A8 core. The SD card slot could be used to store the OS. Angstrom is a Linux based OS which is specifically designed for embedded systems. It has some key features which make it a good choice OS. Since every Beagle bone is shipped with angstrom already present in it, there is no need for additional SD Card storage. Additionally, the angstrom version already has OpenCV modules readily accessible.

The Beagle Bone is a low power open source hardware produced by Texas Instruments. The board was developed by a small team of engineers as an educational board that could be used in colleges around the world to teach open source hardware and open source software capabilities. Beagle Bone is based on an OMAP3530 application processor featuring an ARM Cortex-A8 running at up to 720MHz and delivering over 1,200 Dhrystone MIPS of performance via superscalar operation with highly. The Beagle Bone-xM processor is the DM3730CBP 1GHz version and comes in a .4mm pitch POP package. POP is a technique where the memory is mounted on top of the processor

B. Camera

Logitech C100 camera with auto focus which can process live video with 640*480 resolution at 30 frames per second. The camera is to be placed on the front side of the goggles facing the eye. The eye position is constantly recorded by the camera. The Camera is initialized whenever the Beagle Bone is powered on. On initiation, it initially would capture images and check if any presence of
the eye is detected. If presence is not detected, it would generate a signal to indicate that eye has not been detected.

C. Wireless Relay Module

In order to trigger the various day to day appliances we use relay circuits in the network wherein the inputs are derived from the Beagle Bone. The module consists of an RF Transmitter and Receiver along with voltage level converting circuits and decoder networks. The RF transmitter frequency is 434 MHz and the modulation scheme used is ASK. The transmitter receives inputs from the Beagle Bone which are derived from the eye position. The control signals are received by the receiver module which is placed in the switch board.

IV. SOFTWARE DETAILS

Microsoft Visual Studio is used for developing the eye based tracking code with the help of the OpenCV libraries that are linked to it. A natural choice to make is which programming language the system should be implemented in. OpenCV supports three languages as a default: C, C++ and Python. C++ was chosen since it has support in Visual Studio.

All the relevant Open CV libraries are linked to the Visual Studio and an executable file is generated after compilation. The executable file when executed would begin the eye tracking operation. For appliance control part, the earlier code with little modifications to get control of the GPIO pins the Beagle Bone is used. The binary is built using the OpenCV dlls and the application code for the wireless relay. The binary is triggered each time the system boots up so that it could function independently, thereby making it a plug and play embedded device.

A. Algorithm for Cursor/Appliance Control

1. The root level algorithm to perform image processing for the real-time feed provided by the camera is:
2. The video from the web camera is divided into frames.
3. The required face classifier is loaded and face detection is performed
4. On successful face detection required eye classifier is loaded and eye detection is performed.
5. The region of interest is now set to the detected eye in the image
6. Image pre-processing steps are carried out for pupil detection
7. Pupil is detected with Hough Circles method and the centre of the detected circle is the requires coordinates
8. The coordinates are used to enable appropriate movements with the help of windows functions. For appliance control, depending upon the value obtained above, a unique signal is sent to the wireless relay module.

B. Open CV

Open CV stands for open source computer vision library. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, MATLAB, and other languages. Open CV was designed for computational efficiency with a strong focus on real time applications. OpenCV is written in optimized C and can take advantage of multicore processors. The OpenCV functions that allow us to interact with the operating system, the file system, and hardware such as cameras are collected into a library called HighGUI (which stands for "high-level graphical user interface"). High GUI allows us to open windows, to display images, to read and write graphics-related files (both images and video), and to handle simple mouse, pointer, and keyboard events. OpenCV also has built-in cascades for the implementation of Viola-Jones method of face detection.
The library is full of programming functions which enable its users to create and build sophisticated vision applications. Ideally, OpenCV helps the computer to understand what it reads in, identify it as something which we want to use /detect, or something that is not needed, or regarded as noise.

V. RESULTS

A. Face Detection

Real time face detection was achieved using OpenCV libraries with good accuracy. The captured image was converted to grey scale. A black rectangular enclosure indicates the detected face. The results obtained for face detection is shown in the figure below.

Fig. 3. Face Detection

B. Eye Detection

A white rectangular enclosure indicates the detected eye. Accuracy of eye detection was found to be less when compared to face detection.

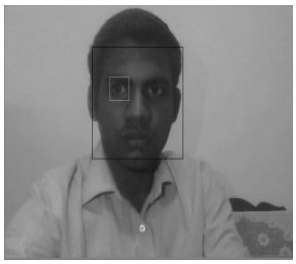

Fig. 4. Eye Detection

C. Wireless Relay

The wireless relay module was successfully implemented. The figure shows the transmitter and receiver part.

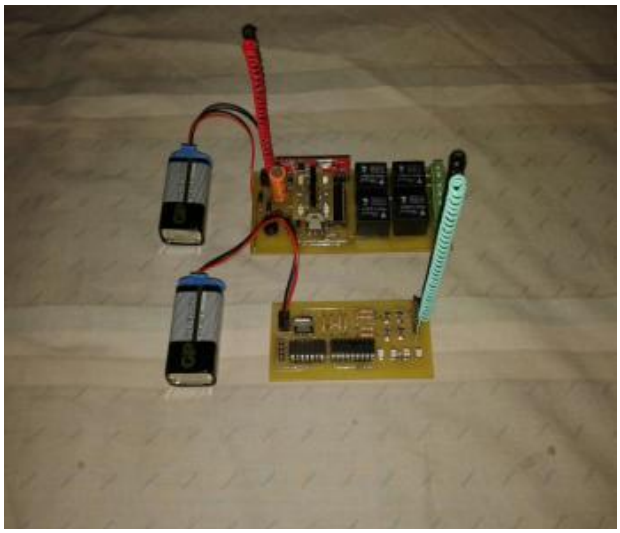

Fig. 5. Wireless Relay

It was found that efficiency of the system depends on the following factors:
1. The distance between the eye and the web camera
Detection algorithms fail to work when the distance from the camera is either too less or too large. So, an approximate distance is set for obtaining efficient results. The distance of the camera is set to be in the range 20-30 cm from the eye.

- Ambient light conditions

Ambient light conditions are necessary for capturing and processing of the image. When the light intensity is too low or too high the processing of image becomes difficult because of the substantial amount of noise involved with each frame and the efficiency was found to decrease.
Due to the factors mentioned above the coordinates obtained tend to vary by a small amount which often affected the calibrated values

So, in order to make the system independent of the coordinate values a method was used where in the difference of the coordinate values between the first and a few frames after the first one was estimated and depending on these values further calibration was done. So irrespective of the frame variations with respect to the user the system worked with better efficiency.

This was because the differential values remained constant even with the variations mentioned above.

The utility of the proposed system is not only pertaining to the applications already mentioned but also it could be extended to various levels from ranging from wheelchair control to locking of enemy targets by fighter pilots, manoeuvring a robot which could be a threat to human life etc. By resolving the co design issues, the system hardware along with the software can prove to be an effective system to make the life of the paraplegic patients independent. The critical part of the system is image processing at real time which can be addressed by using better high end image processing software. The System could be made effective by increasing the sampling rate and resolution of the camera. Having several advantages over other methodologies, it could be having several practical applications including driving assistance in cars, gaming etc.

REFERENCES

1. K.M.Lam, H.Yan,, " Locating and extracting eye in human face images", Pattern Recognition, 29, [5], 771-779, [1996].
2. K.-N. Kim and R. S. Ramakrishna, "Vision-based eye gaze tracking for human computer interface", Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2, 324-329, [1999].
3. Shafi. M, Chung. P. W. H, "A Hybrid Method for Eyes Detection in Facial Images", International Journal of Electrical, Computer, and Systems Engineering, 231-236,[2009].
4. Ito, Nara, "Eye movement measurement by picture taking in and processing via a video capture card, an Institute of Electronics", Information and Communication Engineers Technical Report, 102, 128, 31-36, [2002]
5. Bradski, G. and Kaehler, A. „"Learning OpenCV''. Cambridge, MA: O"Reilly Media, Inc., 2008.
6. Principi E ; Dept. of Inf. Eng., Univ. Politec. delle Marche, Ancona, Italy, Colagiacom V. ; Squartini, S. ; Piazza, F.," Low Power High Performance computing on the BeagleBoard Platfotm", Education and Research Conference, 5° European DSP, 35 - 39 ,[2012]
7. Huang Ying, W. Z.and Xuyan, T. A real-time compensation strategy for non-contact gaze tracking under natural head movement. Chinese Journal of Electroncis (July 2010).
8. Wilson, P. I., and Fernandez, J. Facial feature detection using haar classifiers. J. Comput. Sci. Coll. 21, 4 (Apr. 2006), 127-133.