

Eye Ball Movement to Control Computer Screen

Ramesh R* and Rishikesh M

Department of ECE, College of Engineering Guindy, Anna University, Chennai, Tamilnadu, India

Abstract

The aim of this paper is to present a system for controlling the cursor position in a computer by human eye movements. Many modules have been developed to help the physical world interact with the digital world. Here we present a novel approach for Human Computer Interaction (HCI) where, we control cursor movement using a real time camera and color pointers. Our method is to use a camera and computer vision technology, such as image segmentation, background subtraction and color tracking, to control mouse tasks and we show how it can perform everything as current mouse devices can. A color pointer has been used for the object recognition and tracking, so as to implement the module without any physical contact with the system. The application has been created on matlab environment with operating system as windows 8.1. This method mainly focuses on the use of a web camera to develop a virtual human computer interaction device in a cost effective manner presents hands free interface between computer and human especially for physically disabled persons.

Keywords: Human Computer Interaction (HCI); Eye ball movement; MATLAB; Web camera; Computer

Introduction

As the computer technology is grow up, the importance of human computer interaction is rapidly increasing. Most of the mobile devices and laptops are using touch screen technology. But this technology is still not cheap enough to be used on desktop systems. Creating a virtual human computer interactive module such as mouse or keyboard, can be an alternative way for the touch screen. The motivation is to create an object tracking application to interact with the computer, and develop a virtual human computer interaction device. In our work, we used Matlab to call web camera which is set to take images continuously from the eye focusing pupil. With the help of different image processing techniques, the eye recognition and tracking is achieved. The processing techniques involve an image subtraction algorithm to detect colors. The color is detected from the image pixel and the pixel position is mapped into mouse input position, the root class java program for mouse movement is called to move the cursor [1].

Depending upon the size of the image taken by camera, various scaling techniques are used because the pixel position in the image will not have a correspondence with screen resolution.

Objective

The great challenge lies in developing an economically feasible and hardware independent system so that human can interact without having any physical connection with the computer. The main objective is to develop an object tracking application to interact with the computer, and a virtual human computer interaction device, where in no physical contact with the device is required and also to provide an easier human-machine interaction routine [2].

System approach

- Capturing real time video using Web-Camera focusing pupil of eye.
- Processing the individual image frame.
- Flipping of each image frame.
- Conversion of each frame to a grey scale image.
- Color detection and extraction of the different colors (RGB)

from flipped gray scale image.

- Conversion of the detected image into a binary image.
- Finding the region of the image and calculating its centroid.
- Mouse Movement (Figure 1).

Acquisition

As a primary section, Acquisition is very important. Obtaining a clear image of the pupil in the eye is crucial to the eye-tracking process. A modified low cost USB webcam is used for capturing a close-up picture of one eye pupil. An integral part of the design process, therefore, is to determine the optimum camera settings and light source positions to produce a clear image. Matlab program is used to call the webcam for acquiring the real time video input data.

Using Matlab

The whole system is achieved using a custom matlab script. Matlab is a high -performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar

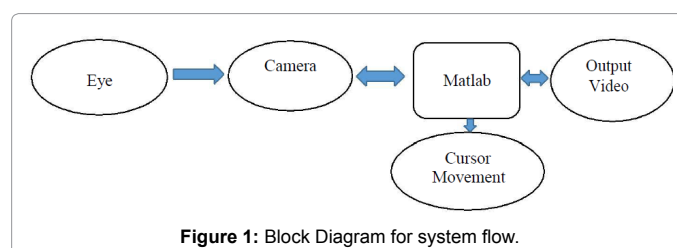


Figure 1: Block Diagram for system flow.

***Corresponding author:** Ramesh R, Department of ECE, College of Engineering Guindy, Anna University, Chennai, Tamilnadu, India, Tel: 04464502506; E-mail: ramesh1arr@gmail.com

Received August 06, 2015; **Accepted** September 22, 2015; **Published** September 30, 2015

Citation: Ramesh R, Rishikesh M (2015) Eye Ball Movement to Control Computer Screen. J Biosens Bioelectron 6: 181. doi:[10.4172/2155-6210.1000181](https://doi.org/10.4172/2155-6210.1000181)

Copyright: © 2015 Ramesh R, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

mathematical notation. It is a powerful tool for image processing. The functions, and the expressions of the matlab language, make image-processing operations easy to write in a compact, clear manner, thus providing an ideal software prototyping environment for the solution of image processing problems (Figure 2).

Image Processing

Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing are also possible. Here image processing is in the sense that we are splitting each pixel of the image into RGB components. This is done by traversing through the image, retrieving each pixel, extracting RGB values and then comparing the color values to the RGB values.

Image analysis

Image analysis is the extraction of meaningful information from images; mainly from digital images by means of digital image processing techniques. Image analysis tasks can be as simple as reading bar coded tags or as sophisticated as identifying a person from their face. In this project we do various image analyzing techniques. The main thing done is the color detection. At first we receive an image from the web cam. Then each pixel is retrieved from the image and extracts the red, green and blue values (RGB) from each pixel. Now we can easily detect a particular color since all the colors are combinations of RGB values. Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions digital image processing may be modeled in the form of multidimensional system [3-5].

Flipping of images

When the camera captures an image, it is inverted. This means that if we move the color pointer towards the left, the image of the pointer moves towards the right and vice-versa. It's similar to an image obtained when we stand in front of a mirror (Left is detected as right and right is detected as left). To avoid this problem we need to vertically

flip the image. The image captured is an RGB image and flipping actions cannot be directly performed on it. So the individual color channels of the image are separated and then they are flipped individually. After flipping the red, blue and green colored channels individually, they are concatenated and a flipped RGB image is obtained.

Steps involved in image processing

Step 1: Capture the video frames using the video input function.

Step 2: Set the properties of video object.

Step 3: Start the video acquisition.

Step 4: Set a loop that starts after 2000 frames of acquisition. This loop contains the following steps:

- Get the snapshot of the current frame.
- Convert the RGB image into grayscale image.
- Now to track the black objects in real time we have to complement the greyscale.
- Convert the resulting complemented grey scale image into a binary image.
- Remove all those pixels which are less than pixels.
- Label all the connected components in the image to perform image analysis, here we get a set of properties for each labelled region.
- Display the image.
- Again a loop is used to bind the black objects in a rectangular box.

Step 5: Stop the video acquisition.

Step 6: Flush all the image data stored in the memory buffer.

Step 7: Clear all the variables.

Step 8: End.

Processing

After the acquisition we have to set a loop that stops after certain frames and the snapshot of current frame is acquired. 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. Grey images requires less time in processing. Then we apply median filter to remove noise from images or frames obtained from the video. The image or frame filtered out with the help of the command 'medfilt2' present in the Image Processing Toolbox.

Feature extraction

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.

Program analysis

The grey region of the image obtained after subtraction needs to be converted to a binary image for finding the region of the detected pupil (Figures 3-5). A grayscale image consists of a matrix containing the values of each pixel. The pixel values lay between the ranges 0 to 255 where 0 represents pure black and 255 represents pure white color. We use a threshold value to convert the image to a binary image. This

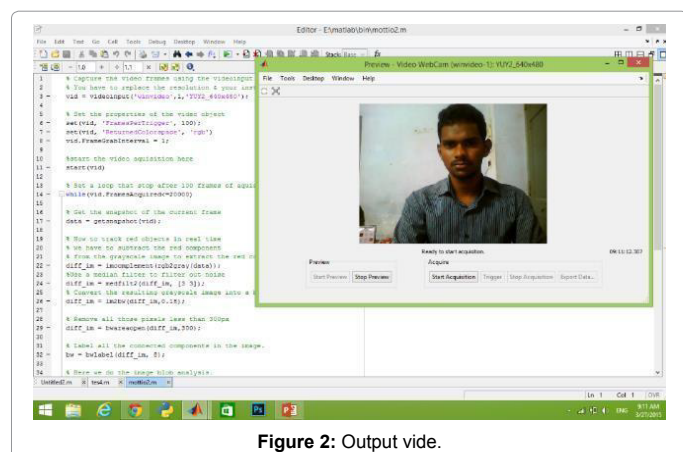


Figure 2: Output vide.

means that all the pixel values lying below threshold value is converted to pure black that is 0 and the rest is converted to white that is thus the resultant image obtained is a monochromatic image consisting of only black and white colors. The conversion to binary is required because MATLAB can only find the properties of a monochromatic image (Figure 6). Final output image: (after boundary detection and centroid): For the user to control the mouse pointer it is necessary to determine a point whose coordinates can be sent to the cursor. With these coordinates, the system can control the cursor movement. An inbuilt function in MATLAB is used to find the centroid of the detected region. The output of function is a matrix consisting of the X (horizontal) and Y (vertical) coordinates of the centroid. These coordinates change with time as the pupil moves. Centroid of the image is detected Coordinates are located and stored in a variable Now the required pupil tracking of eye is achieved (Figure 7). The required Program (*java.awt.Robot) for mouse movement operation is imported in Matlab. The pointer location is determined by assigning X and Y position of bounding box. If the X and Y positions of Bounding box varies, accordingly the position of the cursor will also vary. Now the mouse movement in computer screen is achieved by using pupil of eye movement detection that can be an alternative approach for the touch screen. Works can be done to perform the operations using the inbuilt web camera to track the pupil of eye. This can also be developed as user friendly software by reducing certain complications for which works can be done [5].



Figure 3: Input current frame.

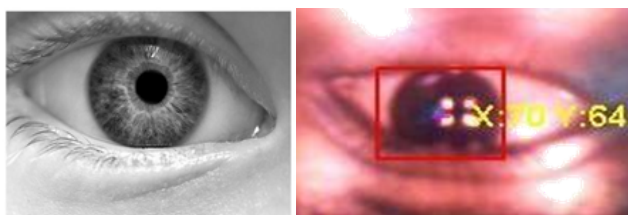


Figure 4: Grayscale image.

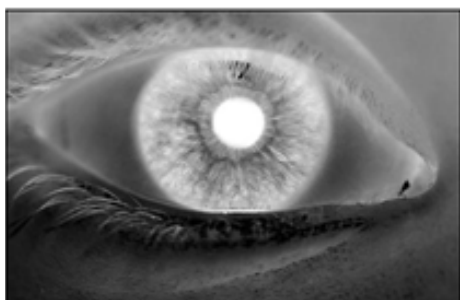


Figure 5: Complemented Grayscale image.

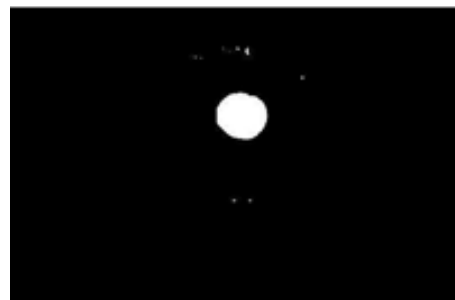


Figure 6: Grayscale image to Binary image.



Figure 7: Head mounted set up.

Results and Conclusion

In this paper, a pupil tracking based computer mouse movement application has been developed and implemented using a webcam. The system has been implemented in Windows 8.1 with 2GB RAM in Matlab environment using Matlab Image Acquisition Processing Toolbox and Image Processing Toolbox is sufficient for the sensitivity of the system to work in real time as how normal mouse perform movement task. Our motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system in more user friendly manner that can be an alternative approach for the touch screen.

Works can be done to perform the operations using the inbuilt web camera to track the pupil of eye. This can also be developed as user friendly software by reducing certain complications for which works can be done.

References

1. Image acquisition Toolbox Functions. Mathworks.
2. Bhargav A (2010) Tracking red color objects using matlab.
3. Solomon C, Breckon T (2010) Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab (3rd edn), John Wiley & Sons.
4. Erdem A, Yardimci E, Atalay Y, Cetin V (2002) Computer vision based mouse. Acoustics, Speech, and Signal Processing, Proceedings. (ICASS). IEEE International Conference.
5. Park H (2008) A Method for Controlling the Mouse Movement using a Real Time Camera. Brown University, Providence RI, USA, Department of computer science, USA.