

Raspberry Pi Based Cursor Movement Using Open CV

K.Manogna Priyadarsini¹, G.Emily Grace²

¹P.G scholar, ²Assistant professor

Dept. of E.C.E, V.R.S & Y.R.N, College of Engg. & Tech., Prakasam District, Chirala, Andhra-Pradesh, India.

Abstract - Some peoples cannot able to operate computers because of an illness. The idea of eye controls of great use to not only the future of natural input but more importantly the handicapped and disabled. Moreover implementing a controlling system in it enables them to operate computer without the help of another person. It is more helpful to handicapped peoples. Those are need to operate computers without hand this one is most useful those can operate cursor by movement of eye. In this paper eye movement sensor through control the cursor. First detect pupil centre position of eye. Then the different variation on pupil position gets different movement of cursor. The Implementation process for Pupil detection using Raspberry pi and Open CV.

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. Rather, a SD/MMC card slot is provided for this purpose. After boot load, as per the application program Raspberry Pi will get execute.

Keywords - Raspberry Pi, Open CV, Eyeball Tracking.

I. INTRODUCTION

Eye movement controlled wheelchair is to enable complexly paralyzed patient to make their life more accessible and to provide them opportunity of independence and movement. However, to steer own wheelchair through a conventional joystick is difficult for people experience total paralysis in all four limbs ,such as muscular dystrophy , spinal cord injury , amyotrophic lateral sclerosis, etc. The idea of eye control is of great use to not only the future of natural input but more importantly the handicapped and disabled. People who are unable to walk and are using wheel chairs exert great amounts of energy using physical strength to turn a steer the wheels. With eyesight Bering their guide, the disabled would save being their guide, the disabled would save energy and could use their hands and arms for other activities. To design a system that detects the image input. To design or create a browser which is used for mainly visibly impaired people. The signals pass the motor driver to interface with the wheelchair itself. The motor driver will control both speed and direction

to enable the wheelchair to move forward, left and right. Camera captures the image. Focus on eye in image by opencv code. Pupil detection will be done by opencv code .Raspberry pi board use in this research. Raspbian OS install the opencv and USB camera configuration. Focus on eye movement and detect the centre position of pupil by eye movement. Take the centre position value of pupil as reference, and then the next the different value of X, Y coordinates will be set for particular command.

II. SYSTEM ARCHITECTURE

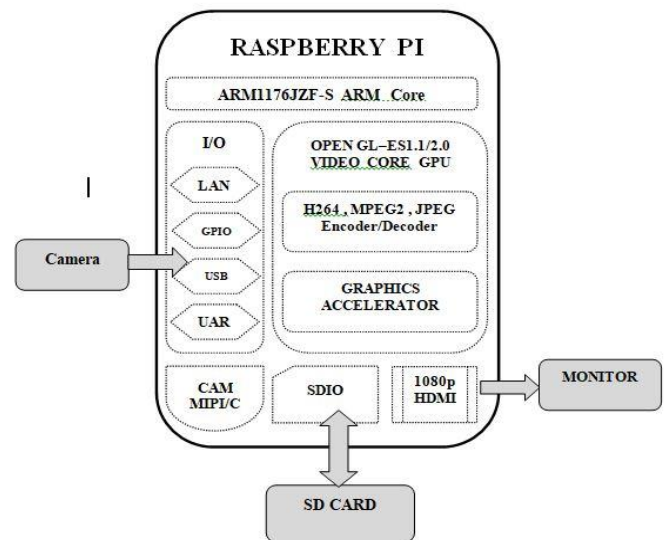


Figure: 1 Block Diagram

III. DESIGN AND IMPLEMENTATION

BRIEF (Binary Robust Independent Elementary Features) - We know SIFT uses 128-dim vector for descriptors. Since it is using floating point numbers, it takes basically 512 bytes. Similarly SURF also takes minimum of 256 bytes (for 64-dim). Creating such a vector for thousands of features takes a lot of memory which are not feasible for resource-constraint applications especially for embedded systems. Larger the memory, longer the time it takes for matching. But all these dimensions may not be needed for actual matching. We can compress it using several methods like PCA, LDA etc. Even other methods like hashing using LSH (Locality Sensitive Hashing) is used to convert these SIFT descriptors in floating point numbers to binary strings.

These binary strings are used to match features using Hamming distance. This provides better speed-up because finding hamming distance is just applying XOR and bit count, which are very fast in modern CPUs with SSE instructions. But here, we need to find the descriptors first, then only we can apply hashing, which doesn't solve our initial problem on memory.

BRIEF comes into picture at this moment. It provides a shortcut to find the binary strings directly without finding descriptors. It takes smoothed image patch and selects a set of $n_d(x,y)$ location pairs in a unique way (explained in paper). Then some pixel intensity comparisons are done on these location pairs. For eg, let first location pairs be P and Q . If $I(p) < I(q)$, then its result is 1, else it is 0. This is applied for all the n_d location pairs to get a n_d -dimensional bitstring.

This n_d can be 128, 256 or 512. OpenCV supports all of these, but by default, it would be 256 (OpenCV represents it in bytes. So the values will be 16, 32 and 64). So once you get this, you can use Hamming Distance to match these descriptors.

One important point is that BRIEF is a feature descriptor, it doesn't provide any method to find the features. So you will have to use any other feature detectors like SIFT, SURF etc. The paper recommends to use CenSurE which is a fast detector and BRIEF works even slightly better for CenSurE points than for SURF points.

FAST Algorithm for Corner Detection - We saw several feature detectors and many of them are really good. But when looking from a real-time application point of view, they are not fast enough. One best example would be SLAM (Simultaneous Localization and Mapping) mobile robot which have limited computational resources. As a solution to this, FAST (Features from Accelerated Segment Test) algorithm was proposed by Edward Rosten and Tom Drummond in their paper "Machine learning for high-speed corner detection" in 2006 (Later revised it in 2010). A basic summary of the algorithm is presented below.

Feature Detection using FAST

1. Select a pixel P in the image which is to be identified as an interest point or not. Let its intensity be I_p .
2. Select appropriate threshold value t .
3. Consider a circle of 16 pixels around the pixel under test. (See the image below)

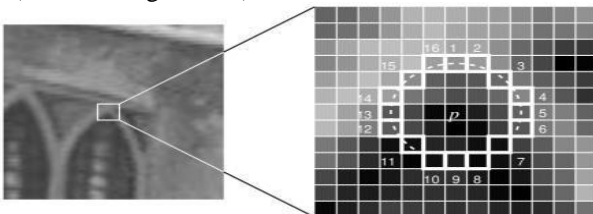


Figure 2: Corner Detection

4. Now the pixel P is a corner if there exists a set of n contiguous pixels in the circle (of 16 pixels) which are all brighter than $I_p + t$, or all darker than $I_p - t$. (Shown as white dash lines in the above image). n was chosen to be 12.
5. A **high-speed test** was proposed to exclude a large number of non-corners. This test examines only the four pixels at 1, 9, 5 and 13 (First 1 and 9 are tested if they are too brighter or darker. If so, then checks 5 and 13). If P is a corner, then at least three of these must all be brighter than $I_p + t$ or darker than $I_p - t$. If neither of these is the case, then P cannot be a corner. The full segment test criterion can then be applied to the passed candidates by examining all pixels in the circle. This detector in itself exhibits high performance, but there are several weaknesses:
 - (i) It does not reject as many candidates for $n < 12$.
 - (ii) The choice of pixels is not optimal because its efficiency depends on ordering of the questions and distribution of corner appearances.
 - (iii) Results of high-speed tests are thrown away.
 - (iv) Multiple features are detected adjacent to one another.

First 3 points are addressed with a machine learning approach. Last one is addressed using non-maximal suppression.

IV. SYSTEM HARDWARE

A Raspberry Pi is a thirty five dollar, credit card sized computer board which when plugged into an LCD and attachment of a keyboard and a mouse, it is able to complete the functions of any regular PC can. Like a PC, it has RAM, Hard Drive (SD Card), Audio and Video ports, USB port, HDMI port, and Ethernet port. With the Pi, users can create spread sheets, word-processing, browse the internet, play high definition video and much more. It was designed to be a cost friendly computer for users who needed one. There are two models, Model A and B. Model B is the faster containing 1GB of RAM as well as the ability to over clock.

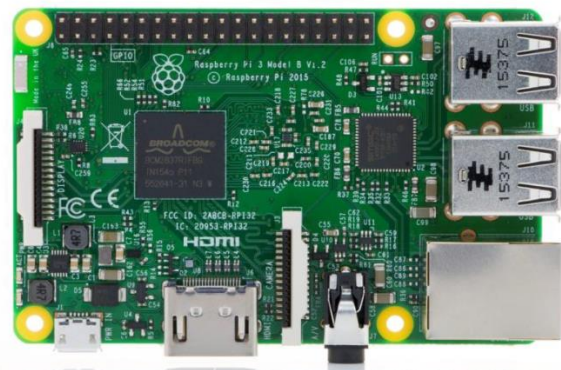


Figure 3: Raspberry Pi Microcontroller

V. SOFTWARE TOOLS

Raspbian Wheezy - Raspbian Wheezy is a free operating system based on Debian distribution. It is created by a small team of developers who are fans of Raspberry Pi. Raspbian is optimized for the Raspberry Pi's hardware and it comes with over 35 000 packages and pre-compiled software. Raspbian is still under active development and it aims to improve the stability and performance of the Debian packages. Raspbian is officially recommended for beginners and it includes the graphical desktop environment called LXDE. Raspbian Wheezy is one of the fastest ways to setup and get the RasPi running.

Programming languages - There are considerable numbers of programming languages which have been adapted for Raspberry Pi. Python programming language is recommended by The Raspberry Pi foundation especially for the beginners. Basically any programming language which can be compiled for ARMv6 can run on the Raspberry Pi. There-fore the users are not restricted to use only the Python. On the Raspberry Pi there are preinstalled several languages for example C, C++, Java, Scratch and Ruby.

Python programming language - Python programming language is developed in the late 1980s at the National Research Institute by Guido van Rossum. Python has grown in popularity, and it is widely used commercially.

Python is a flexible and powerful programming language but still it is easy to learn and follow. The clear syntax of Python makes it a valuable tool for users who wants to learn programming. This is one of the reasons why it is recommended by the Raspberry Pi Foundation. Python is published under an open-source license and it is available for different operating systems. Python runs on Linux, OS X and Windows computer systems.

Cross-platform support guarantees that the programs which are written in Python are also compatible in other platforms. There are few exceptions where the pro-grams are not compatible. For instance, when the Python is addressed to use the specific hardware such like Raspberry Pi's GPIO.

VI. CONCLUSION

In this work, we tested the effectiveness of pointing and scrolling using eye blink sensor on wireless device interfaces. The results indicate that pointing and scrolling can be effectively done using tilting. Fits' law is found to fit the experimental data for both of the tasks but with higher coefficient of determination, R², in the case of scrolling. The results also showed that wrist tilting is relatively easier around the thumb than along it. We think that tilting interaction provides an alternative way of interaction that needs only one hand rather than both hands compared to using the stylus. We noted that users prefer tilting using their non dominant hand, which make the dominant hand free for handling the environment. The result introduced in this work can help in

the design of device interfaces especially when only one hand is available for the interaction.

VII. REFERENCES

- [1]. S. Tameemsultana and N. Kali Saranya, "Implementation of Head and Finger Movement Based Automatic Wheel Chair", Bonfring International Journal of Power Systems and Integrated Circuits, vol. 1, Special Issue, pp 48-51, December 2011.
- [2]. Manuel Mazo, Francisco J. Rodriguez, Jose L. Lazaro, Jesus Urena, Juan C. Garcia, Enrique Santiso, Pedro Revenga and J. Jesus Garcia, "Wheelchair for Physically Disabled People with Voice, Ultrasonic and Infrared Sensor Control ", Autonomous Robots, vol.2, no. 3, pp. 203-224 ,Sep 1995.
- [3]. Tabasum Shaikh, Naseem Farheen Sayyed, Shaheen Pathan, "Review of Multilevel Controlled Wheelchair", 4th National Conference On Electronic Technologies, pp. 275-279, April 2013.