

EyeBall Movement Based Cursor Using Raspberry Pi

Anto rose*, Mirunalini M, Divya darshini N

Department of Computer Science and Engineering,

Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai – 62

***Corresponding author: E-Mail: antorose@velhightech.com**

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 Camera is capturing the image of eye movement. First detect pupil centre position of eye. Then the different variation on pupil position get different movement of cursor. The Implementation process for Pupil detection using Raspberry pi and on the terminal of Raspberry image installed on raspberry pi. The Raspberry Pi is a SOC uses ARM1176JZF-S core. SOC, or System on a Chip, is a method of placing all necessary for running a computer on a chip. Raspberry Pi needs an OS to start up. In the aim of cost reduction, the Raspberry Pi omits any on-board non-volatile memory used in embedded systems. SD card slot is provided for this purpose. After boot the application program Raspberry Pi will get execute.

KEY WORDS: Raspberry pi, Camera, Open C, Pupil, SDcard, Linux, system-on-chip, eyeball, display.

1. INTRODUCTION

Embedded system is a combination of hardware and software components. Embedded system has two separate processor and controller. Processor doesn't have any separate memory it is just a temporary storage but controller has its own memory for storage purpose. Controller can be used instead of processor. Controller is inbuilt chip. In computing there are two types they are CISC (complex Infrastructure Set Computing), RISC (Reduced Infrastructure Set Computing). CISC is slower when compared to RISC. Raspberry Pi is a controller it consist of camera module, audio module, HDMI (High Definition Multimedia Interface) the pc is connected via HDMI port, separate power cable, display, Ethernet cable and two USB ports where the mouse, keyboard will be connected. Raspberry pi can be used instead of CPU. CPU consist of operating system for storage purpose, but the Raspberry pi controller doesn't have any operating system in it. But the Raspberry pi has separate SD card attachment for storage purpose. SD storage capacity is about 64 GB. Pendrive can also be used for storing, the total storage capacity of pendrive is about 2 TB. ARM 11 controller can also be used instead of Raspberry pi. Since Raspberry pi controller and ARM11 controller uses the python language. ARM is defined as Advanced RISC machine RISC follows only the bus topology format. Open C is a software which will be used for matching the position of the eyeball. Open C contains the database which consist the original eyeball position through which the matching is done. Raspberry pi 3B advanced version controller is used. The camera module captures every movement of eye. It controls cursor-moving by automatically affecting the position where eyesight focus on, and simulates mouse-click by blinking action. However, the proposed vision-based virtual interface controls system work on various eye movements such as eye blinking. 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 the disabled. Moreover implementing a controlling system in it enables them to operate computer without the help of another person. Computer programming's importance in industry has placed increasing demands on computer programming education. Many instructors and researchers have investigated how to effectively develop students' programming skills. However, programming is still challenging for most novices, particularly in the area of debugging, where programmers have to locate and resolve bugs in programs. Because debugging is an essential skill in programming, students identified as proficient in debugging are typically expert in programming. Therefore, many studies have explored program debugging by comparing expert and novice programmers to identify the factors influencing their programming/debugging performance. To comprehensively investigate programming/debugging, research has explored students' cognition during programming/debugging. Traditional research methods, such as questionnaire studies, interviews, and observation, usually explain cognition in a more subjective manner. Although "think-aloud" can reveal a more authentic cognitive process, it involves a high cognitive load. To provide insight based on more solid evidence, recent research has started to investigate human cognition using cognitive neuroscience techniques, such as the eye-tracking method. The aim of the study reported in this paper was to depict students' cognitive processes during program debugging by comparing high- and low-performance students based on the eye-tracking method, so that effective and adaptive instructional strategies can be developed to improve students' understanding of programming. The contributions of "Temporal" and "sequential" information related to eye gaze data during debugging was shown to reveal a dynamic and nonlinear procedure for debugging. The dynamic process of eye gaze was used to derive more insight into cognitive processes. By integrating proportional statistics, sequential analysis, and interviews, solid evidence was provided for the validity of existing theories of program comprehension and

debugging. The personal computer systems are carrying a huge part in our daily lives as they are used in areas such as work, education and enjoyment. All these applications have in common is that the use of personal computers is mostly depend on the input method through 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. By exploring the possible causes for the challenges experienced by programmers, the research findings confirm existing theories of program comprehension and debugging strategies, and support existing findings on the differences between experts and novice.

2. RELATED WORK

Most of the debugging from the perspective of students' exterior behaviour, that is, investigating students' actions (e.g., how students write programs or answer a questionnaire) instead of investigating students' inner cognitive process during debugging (e.g., how students comprehend programs based on their mental models). Cognition comprises the mental activities and processes based on an individual's psychological functions, reflecting the higher-level functions of the brain. Mapping the cognitive process, therefore, can help researchers or instructors to understand the psychological causes of behaviour. Traditional research has studied cognition by using interviews, behavioural observation, or questionnaires. These subjective methods depend heavily, however, on the subjective opinions to complex cognitive processes. To avoid subjectivity, verbal protocol analysis, for example "think-aloud," is a more accurate method investigating cognitive processes. Some research has applied verbal protocol analysis to explore the cognitive processes during programming. This was based on verbal protocol analysis, which has its own mental states. However, this method still has some limitations. Subjects can only verbalize thoughts and processes of which they are consciously aware, and thinking aloud creates extra cognitive load. Eye movements loaded by eye trackers can have much information on cognitive processes, have been used to investigate problem-solving processes, learning and reading. Some research investigated the differences between experts based on eye-tracking analysis and suggest that the allocation can be influenced by expertise. Through the process of eye tracking, it is possible to obtain visual attention information such as eye fixation (pauses over informative regions of interest), saccade (rapid movements between fixations), gaze duration (the total duration of all fixations on a region prior to moving to another region), and reification (fixation on information that has already been fixated on) in a Region Of Interest (ROI); this can help for cognition activities that occur continuously. The ROI is an image labelled based on a particular purpose. Bednarik classified ROIs according to the visualization components of the visualized programming integrated development environment (IDE), including the source code area, the visualization area, and output area, to observe the subjects switched their attention between different visual areas. In this cases it would be preferable to use input methods which are based on our abilities of the region such as the eye movements. Enabling such substitute input methods a system was made that follows a low-price approach to control a cursor on a computer system. The eye tracker is based on various images recorded by a mutated webcam to acquire the various eye movements. These eye movements are then graphed to a computer screen for positioning a cursor accordingly. The movement of mouse can be grabbed automatically by adjusting the position of our eyesight. Camera is used for capturing the image of eye movement. Now there has been a growing interest in developing natural interaction between human and the computer. There are several studies for human-computer interaction in universal computing. There is a vision-based interface technique that extracts motion without any high cost equipments from an input video image. So that the vision-based approach is taken into account for an effective technique to develop the human computer interface systems. For vision-based human computer interaction, eye tracking is an issue. Real time eye input has been used most frequently used for disabled users, who can use only their eye movements as input. This explores students' cognitive processes while debugging the programs by using an eye tracker. The path of students' while following program codes was subjected to the sequential analysis to reveal significant sequences of areas. These significant path sequences were compared to those students with different debugging performances. Low-performance students can often jump directly to certain statements to find bugs, without the program's logic. They were often needed to trace prior statements to recall information. Based on the results, adaptive instructional strategies and materials can also be developed for students by different performance levels, to improve associated cognitive activities while debugging, which can be learnt during debugging and programming. Making the machine to the user is one of the aspects of effective computing. Signal process by Brain Computer Interface frameworks has been utilized to process brain signals for controlling different gadgets for human ease. P300 based content speller is the device utilized for eye-writing purpose, eye movement based content speller can able to create fine and exact about contrasted with P300 content speller. This paper presents the robust solution for utilizing eye muscular-movement for typing the text content.

Literature Survey:

Driver Distraction Detection by Semi-Supervised Machine Learning: Real-time driver distraction detection is a core to many distraction countermeasures and fundamental for constructing the driver assistance system. While data driven methods demonstrates detection performance, a particular challenge is to reduce the considerable cost for collecting mentioned data. The paper explored semi-supervised methods in driver distraction detection in driving

conditions to alleviate the cost of mentioned training data. Some support vector machine and semi-supervised extreme learning machine were evaluated using the eye and head movements to classify driver states attentive and cognitively distracted. With additional unmentioned data, the semi-supervised methods improved the detection performance (G -mean) by 0.0245, on an average, over all subjects, as compared with the supervised methods. As unmentioned training data can be collected from drivers' natural driving records with some extra resource, semi-supervised methods, which utilize both labelled and unlabeled data, can enhance the efficiency of development in terms of the time and cost.

Forecasting Next Shot Location in Tennis by Fine-Grained Spatiotemporal Tracking Data: In most of the professional sport, an enormous amount of the fine-grain performance data can be generated at the near millisecond intervals in the form of vision-based tracking the data. One of the first sports to this technology has tennis, where Hawk-Eye technology has been used to aid umpiring decisions, and for visualizing shot trajectories for broadcasting purposes. Despite the high-level of accuracy of tracking systems and the sheer volume of spatiotemporal data generates, the use of this data for player performance analysis and their prediction has been lacking. In this, we use ball and player tracking data from "Hawk-Eye" to discover the unique player styles and to predict within-point events. We move beyond the current analysis that only incorporates coarse match statistics and use spatial, temporal information with better characterizes the tactics and tendencies of player. Using a probabilistic graph model, we are able to model player behaviours which enables us to find the factors such as location and speed of the incoming shot which are most conducive to a player hitting the winner (i.e. "sweet-spot") or cause an error, and do "live in-point" prediction - based on the shots played by a rally to estimate the probability of the outcome (e.g. winner, continuation or error) and the location of next shot. As player behaviour based on the opponent, we use this model adaptation to enhance our own prediction. We show the utility by analyzing the play of Federer at 2012 Australian Tennis Open.

Novel Eye-Gaze-Controlled Wheelchair System for Navigating Unknown Environments: In this electric wheelchair design, person with motor impairments due to the diseases, such as amyotrophic lateral sclerosis (ALS), have tools to form more independent and mobile. However, an electric wheelchair requires considerable skill to learn how to use them and operate. Moreover, some people with motor disabilities that cannot drive an electric wheelchair manually (even with a joystick), because they lack the physical ability to control the hand movement (such is the case with people with ALS). In this paper, we have a novel system that enables a person with motor disability to control a wheelchair through eye-gaze and to provide continuous, real-time navigation with unknown environments. The system consists a Per mobile M400 wheelchair, eye tracking glasses, a depth camera to capture the geometry of ambient space, a set of ultrasound and infrared sensors to detect the obstacles with low proximity that are out of view for the depth camera, a laptop placed on a flexible mount for maximized comfort, and a safety off switch to turn off the system whenever it is needed. First, a novel algorithm is proposed for support continuous, real-time target identification, path planning, and navigation with unknown environments. Second, the system utilizes a novel N-cell grid-based graphical user interface that is used to adapt input/output interfaces specifications. Third, a calibration method for eye tracking system is implemented for minimizing the calibration overheads. A case study with a person with ALS is presented, and interesting things are discussed. The participants showed improved performance in terms of the calibration time, task completion time, and navigation speed for navigation trips between them, dining room, and bedroom. Furthermore, the caregiver has also shown promising results such as the participant enjoyed higher level of condense driving the wheelchair and experienced no collisions via all the experiment.

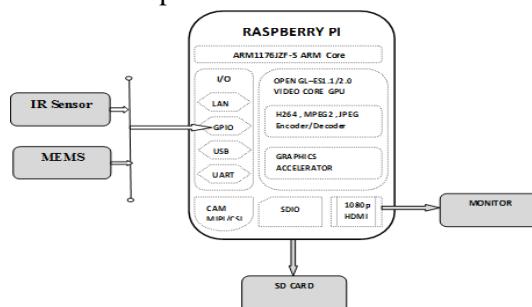


Figure.1. Block Diagram

Block diagram description: Here, in this block diagram the whole system is controlled by Arm11 processor and this processor is implemented on Raspberry Pi Board. so this board is connected with monitor, camera, and SD card. Those all components are connected by USB adaptors. Raspberry pi is the key element in processing module which keeps on monitors eye movement by interfacing USB camera. Camera is capturing the image of eye movement. USB 2.0 makes USB Cameras ideal for many imaging applications. USB Camera will be interface with raspberry pi. Raspberry pi will be use SD card, then the install raspbian OS and open C on raspberry pi. First image will be capture by USB Camera. Focus on eye in image and detect the centre position of pupil by open C code. Take the centre position value of pupil as reference, and then next the different value of X, Coordinates will be set for particular command. Then it is transmitted through zigbee.

Raspberry Pi: Raspberry Pi board is the miniature marvel, packing considerable computing power into the footprint that is not larger than a credit card. It's capable of some amazing things, but there are few things that we are going to know before we plunge head-first into the bramble patch.

Connecting to a Display: Before we start using our Raspberry Pi, we are going to connect a display. The Raspberry Pi supports three different video outputs. Composite video and HDMI video are accessible by the end user, in this section, the DSI video requires some specialized hardware.

Composite Video: Composite video, available via the yellow-and-silver port at the top of the Pi known as an RCA phone connector, is designed for connecting the Raspberry Pi to other display devices. The connector creates a composite colors found within an image—red, green and blue—and sends it down a single wire to display device, typically an old cathode-ray tube (CRT) TV. Figure.1. The yellow RCA phone connector, is used for composite video output. When no other display device is available, a composite video connection will get started along with the Raspberry Pi. The quality, isn't great. video connections are significantly more prone to the interference, more of blurriness and run at a limited resolution, which means that we can fit fewer icons and lines of the text on screen at once.

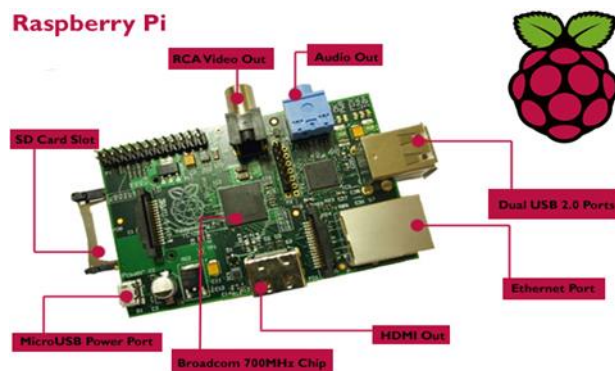


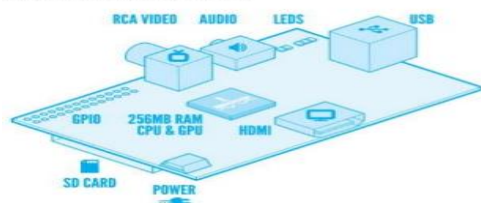
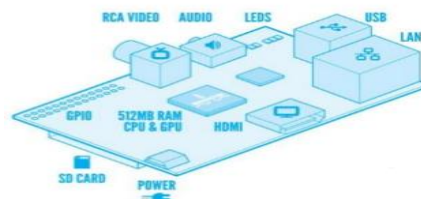
Figure.2. Raspberry Pi Model

HDMI Video: A high quality picture can be obtained using the HDMI (High Definition Multimedia Interface) connector, the only port found at the bottom of the Raspberry Pi (Figure.2). Unlike the composite connection, the HDMI port provides the high-speed digital connection for the pixel-perfect pictures on both computer monitors and the high-definition TV sets. Using this HDMI port, a Raspberry Pi can display images at the Full HD format 1920x1080 resolution of the most modern HDT. If we are hoping to use the Raspberry Pi with an existing computer screen, we may find that our display doesn't have an HDMI input. The digital waves present on the HDMI cable to a common computer screen standard form called DVI (Digital Video Interconnect). By purchasing an HDMI-to-DVI cable, we will be able to connect the Raspberry Pi's HDMI port to the monitor with DVI-D connectivity. The silver HDMI connector, for the high-definition video output. If our monitor has a VGA input—a D-shaped connector with 15 pins, typically colored silver and blue—the Raspberry Pi can't get connected to it. Adapters are available that will take in a digital DVI signal and then convert it to an analogue VGA signal, but these are expensive and bulky. The best option here is simple to buy a modern monitor with a DVI or HDMI input.

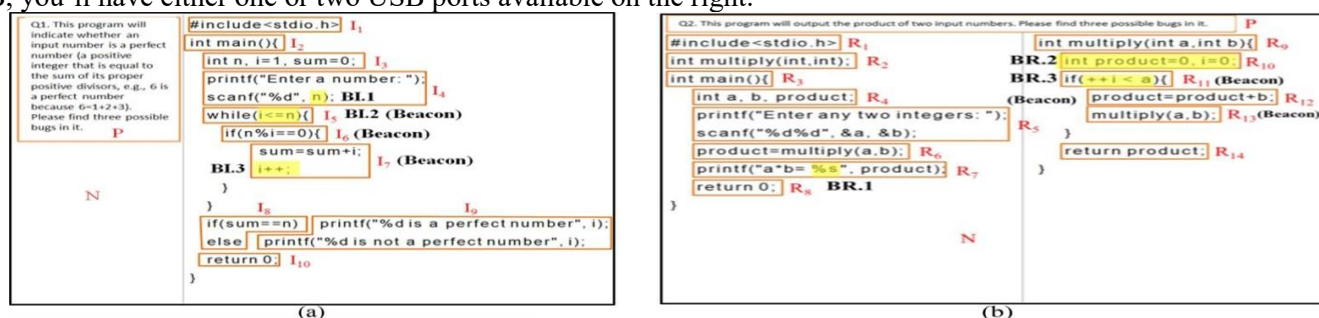
DSI Video: The final video output on the Pi can be found on the SD card slot that is on the top of the printed circuit board—it's a small ribbon connector protected by the layer of plastic. This is for a video standard form known as Display Serial Interface (DSI), which is used in the flat-panel display such as tablets and smart phones. To view with a DSI connector are available for retail purchase, and are typically reserved for engineers whose who are looking to create a compact, self-contained system. A DSI display can be connected to the Pi by inserting a ribbon cable into the matched connector on the Pi, but for beginners, the use of a composite or HDMI display is mostly recommended.

Connecting Audio: The HDMI port carries both the video signal and the digital audio signal. This means that we can connect a single cable to our display device to enjoy both sound and picture. Assuming that we are connecting the Pi to a standard HDMI display, there is very little for doing it as point. It's enough to simply connect the cable. If we are using the Raspberry Pi with the DVI-D monitor through an adapter or cable, audio also will not be included. This highlights the difference between HDMI and DVI: while HDMI can carry out the audio signals, DVI cannot. For those with DVI-D monitors, or those by the composite video output, a black 3.5 mm audio jack located on the top edge of the Raspberry Pi next to the yellow phone connector provides the audio. This is same connector and also used for headphones and microphones on the consumer audio equipment, and it's wired in exactly the same way. If we want, we can simply connect the pair of headphones to this port for quick access of audio. While headphones are connected directly to the Raspberry Pi, we may find the volume that is little lacking. If possible, connect a pair of powered speakers instead of the amplifier inside will help to boost the signal to a more audible level. If we are looking for something that is more permanent, we can either use standard PC speakers that have a 3.5 mm connector or we can buy some adapter cables. For composite video users, a 3.5 mm to RCA phone cable is very useful. This provides a two white and- red RCA phone connections that has a long side video connection, each carrying the

channel of a stereo audio signal to the TV. we have on our system. Similar cable types are readily at consumer electronics shops, or that can be purchased even cheaper at the online retailers such as Amazon.

RASPBERRY PI MODEL A**RASPBERRY PI MODEL B****Figure.3. Raspberry Pi Model A, B**

Connecting the Keyboard and Mouse: Now we have got our Raspberry Pi's output devices sorted, it has time to think about input. As a bare minimum, we are going to need a keyboard, and for the majority of users, a mouse or trackball is the necessity too. First, some news: if we have got a keyboard and mouse with a PS/2 connector—a round plug with the horseshoe-shaped array of pins—then we are going to have a replacement. The old PS/2 connection has been superseded, and the Raspberry Pi side of the Pi (Figure.3.). If you're using Model B, we can connect the keyboard and mouse directly to these ports. If we are using Model A, we will need to purchase a USB hub in order to connect two USB devices simultaneously. Figure.3. Model B's two USB port A USB hub is a good investment for any Raspberry Pi user: even if we have got a Model B, we should use up both our available ports just connecting our keyboard and mouse, leaving nothing free for additional devices such as an external drive, storage device or joystick. Make sure to buy a powered USB hub: models that are smaller, but lack in the ability to run current hungry devices like CD drives and hard drives. If we want to reduce the number of power sockets in use, connect the Raspberry Pi's USB power lead to the powered USB hub. This way, the Raspberry Pi can draw its power directly from the hub, rather than needing its own dedicated power socket and its mains adapter. This will only work on hubs with a power supply capable of providing 700mA to the Raspberry Pi's USB port, along with power is required by other peripherals. Connecting the keyboard and mouse is as simple as plugging them in to the USB ports, or directly in the case of a Model B or through a USB hub in the case of a Model A. A Note on Storage As we have probably noticed, that Raspberry Pi doesn't have a traditional hard drive. Instead of that it uses a Secure Digital (SD) memory card, a solid-state storage system typically used in the digital camera. Almost any SD card will work with the Raspberry Pi, but it holds the entire operating system, it is necessary for the card to be atleast 2 GB in capacity to store all the required files. SD cards with the operating system preloaded that are available from the official Raspberry Pi Store along with the numerous other sites on the Internet. If we have purchased one of these, or received it in a bundle with your Raspberry Pi, we can simply plug it to the SD card slot at the bottom side of left-hand edge. If not, we need to install an operating system—known as flashing—onto the card before it is ready to go. Some SD cards work better than the others, with some models refusing to work at all the Raspberry Pi. For an up-to-date list of SD card models known to work with the Raspberry Pi, visits the eLinux Wiki page. Expects our peripherals to be connected over the Universal Serial Bus (USB) port. Depending on this we should purchase the Model A or Model B, you'll have either one or two USB ports available on the right.

**Figure.4. ROIs for two test programs: (a) the iterative program and (b) the recursive program**

An Introduction to Python: The Raspberry Pi is a fruit name to new computing systems—from classic micro computers like the Acorn, Apricot and Tangerine to more recognizing modern brands including Apple and BlackBerry—but the secondly comes courtesy of the Python programming language. Flexible and powerful, Python was originally developed in late 1980s at the National Research Institute for Maths and Computer Science by Guido van Rossum as a successor to the ABC language. Since its introduction, Python has grown in world wide to what is seen as a easy syntax developed with a focus on ensuring that the code is readable. Python is the high-level language. This means that Python code is written in largely recognizable English, providing the Raspberry 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 is closer to the computer “thinks” but almost impossible for a human to follow it without experience. The high-level nature and clear syntax of Python makes it a valuable tool for anyone who wants to learn the program. It is also recommended by the Raspberry Pi Foundation for those looking in progress to form the simple Scratch.

3. SIMULATION RESULTS

Raspberrypi is clubbed with USB Camera. Raspberry pi will be used as SD card, then install raspbian OS and open cv on raspberry pi. First image is captured by USB Camera. Focus on eye in the image and detect the center position of the pupil by opencv code. Take the exact position value of pupil as reference, and then different value of X, Y coordinates will be set for accurate command. Raspberry pi signals are passed by GPIO to transistor circuit. One main aim for Eye Movement controlled cursor is to enable handicap persons to make their life more feasible and to provide them capability and opportunity of independence. The pupil to control the cursor on the screen, the central coordinate of the screen is set as starting point.

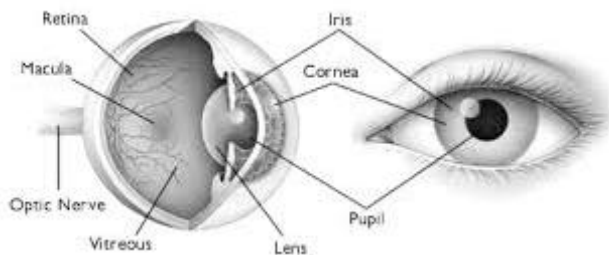


Figure.5. Pupil anatomy

This position is used as base for tracing, and First position of the mouse pointer is set as the center of the screen. The moving position of cursor takes the initial position as base. As the pupil moves to one direction, the coordinate of the mouse pointer on screen change as per the action of pupil. When pupils return to the original position, cursor stops moving. The starting point and stopping point of the cursor is also controlled by movement of the eyes. When the user moves his eyes for a second, then the cursor either starts or stops moving i.e, if the cursor is in moving condition, then after a eye movement, it stops and vice versa. The eye direction arrows are used for testing at the time of demonstration. Beside these arrows, circle is seen in green color. When any obstacle is detected in front of the eyes, this circle becomes red, a buzzer beeps and the cursor stops. This circle does not appear red for natural stopping of the cursor by eye movement. The eye movement and its circuitry required two 1.5 V batteries, three 9V batteries. The camera used, is USB camera which can be straightly plugged in and can be used. The camera drivers should be loaded in the system before using.

4. CONCLUSION

This paper is aim for the cursor control using eye ball movement. Without using the hands we can operate the computers by eye movement. This technology is contracted to replace the conventional computer screen pointing device for the use of physically handicapped persons. The movement of mouse pointer by automatically adjusting the position of eyesight. Without the help of other person they can use the computers and work in it. This work can be extended to implement to produce efficient movements to perform the click events and also cover the total mouse function of the system and the total human-computer interface using eye blink. Technology also extended to the eyeball movement and eye blinking to get efficient and accurate result.

Future Enhancement: Human Computer Interfaces has more scope in upcoming days. There are three aspects as far as future enhancements are concerned.

- Making of user friendly applications by which one can paint, chat and play virtual games.
- Use of other areas of scalp for EOG acquisition.
- Make a sophisticated product that can be implemented in fighter jets. Pilot just has to point the target with the eyes and press the trigger when target becomes under the range. It is very nice that the eye movement is more fast and accurate than just the hand movement.

REFERENCES

- Ahmadzadeh M, Elliman D and Higgins C, An analysis of patterns of debugging among novice computer science students, ACM SIGCSE Bull, 37 (3), 2005, 84.
- Bednarik R and Markku T, An eye-tracking methodology for characterizing program comprehension processes, in Proc. ACM Symp. Eye Tracking Res. Appl, 2006, 125–132
- Bednarik R and Markku T, Temporal eye-tracking data, Evolution of debugging strategies with multiple representations, in Proc. ACM Symp. Eye Tracking Res. Appl, 2008, 99–102.
- Bednarik R, Busjahn T and Schulte C, Eye movements in program-ming education, Analyzing the expert's gaze, in Proc. 1st Int. Work-shop, University of Eastern Finland, 2014, 16–19,
- Bednarik R, Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations, Int. Human-Comput. Studies, 70 (2), 2012, 143–155.

Branch J.L, Investigating the information-seeking processes of ado-lescents, *Libr. Inf. Sci. Res*, 22 (4), 2000, 371–392.

Busjahn T, Eye movements in code reading?, Relaxing the linear order, in *Proc. ICPC*, Florence, Italy, May 18–19, 2015, 255–265.

Busjahn T, Schulte C and Busjahn A, Analysis of code reading to gain more insight in program comprehension, in *Proc. 11th ACM Koli Calling Int. Conf. Comput. Educ. Res*, Koli, Finland, 2011, 1–9.

Crosby M, Scholtz J and Wiedenbeck S, The roles beacons play in comprehension for novice and expert programmers, in *Proc. 14th Annu. Workshop Psychol. Program. Interest Group*, London, U.K, 2002, 58–73.

Davies S.P, Models and theories of programming strategy, *Int. J. Man-Mach. Studies*, 39, 1993, 237–267.

Fan Q, The effects of beacons, comments and tasks on program comprehension process in software maintenance, *Pro Quest LLC*, Ann Arbor, MI, USA, 2010.

Fitzgerald S, Debugging, Finding, fixing and flailing, a multi-institutional study of novice debuggers, *Comput. Sci. Educ*, 18, 2008, 93–116.

Jadud M.C, A first kook at novice compilation behaviour using BlueJ, *Comput. Sci. Educ*, 15, 2005, 25–40.

Jarodzka H, Scheiter K, Gerjets P and Van Gog T, In the eyes of the beholder, How experts and novices interpret dynamic stimuli, *Learning Instruct*, 20, 2010, 146–154, 2010.

Lai M.L, A review of using eye-tracking technology in ex-ploring learning from 2000 to 2012, *Educ. Res. Rev*, 10 (88), 2013, 90–115.

McGill T and Volet S, A conceptual framework for analysing stu-dents' knowledge of programming, *J. Res. Comput. Educ*, 29 (3), 1997, 276–297.

Milne and Rowe G, Difficulties in learning and teaching program-ming—views of students and tutors, *Educ. Inf. Technol*, 7, 2002, 55–66.

Murphy L, Debugging, The good, the bad, and the quirky—A qualitative analysis of novices' strategies, *SIGCSE Bull*, 40 (1), 2008, 163–167.

O'Brien M.P and Buckley J, Inference-based and expectation-based processing in program comprehension, in *Proc. 9th IWPC*, 2001, 71–78.

Perkins D.N, Hancock C, Hobbs R, Martin F and Simmons R, Conditions of learning in novice programmers, *J. Educ. Comput. Res*, 2, 1986, 37–55.

Renumol V, Classification of cognitive difficulties of students to learn computer programming, *Indian Inst. Technol*, 2009.

Renumol V.G, Janakiram D and Jayaprakash S, Identification of cognitive processes of effective and ineffective students during com-puter programming, *Trans. Comput. Educ*, 10 (3), 2010, 1–21.

Sharif B, Falcone M and Maletic J.I, An eye-tracking study on the role of scan time in finding source code defects, in *Proc. ACM Symp. Eye Tracking Res. Appl*, 2012, 381–384.

Simon B, Common sense computing (episode 4), Debugging, *Comput. Sci. Educ*, 18 (2), 2008, 117–133.

Tsai M.J, Hou H.T, Lai M.L, Liu W.Y and Yang F.Y, Visual attention for solving multiple-choice science problem, An eye-tracking analysis, *Comput. Educ*, 58, 2011, 375–385.

Uwano H, Nakamura M, Monden A and Matsumoto K.I, Ana-lyzing individual performance of source code review using reviewers' eye movement, in *Proc. ACM Symp. Eye Tracking Res. Appl*, 2006, 133–140.

Van Gog T, Kester L, Nievelstein F, Giesbers B and Paas F, Un-covering cognitive processes, Different techniques that can contribute to cognitive load research and instruction, *Comput. Human Behav*, 25 (2), 2009, 325–331.

Verdú E, A distributed system for learning programming on-line, *Comput. Educ*, 58 (1), 2012, 1–10.

Villalobos J.A and Calderon N.A, Developing programming skills by using interactive learning objects, in *Proc. ITiCSE*, 2009, 151–155

Yu-Tzu Lin, Cheng-Chih Wu, Ting-Yun Hou, Yu-Chih Lin, Fang-Ying Yang, and Chia-Hu Chang, Tracking Students' Cognitive Processes During Program Debugging—An Eye-Movement Approach, 59 (3), 2016.