IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727 Volume 7, Issue 2 (Nov-Dec. 2012), PP 23-28
www.iosrjournals.org

Eagle i-Bot: An Eye-controlled System

Mohona Gazi Meem [1], Onindita Afrin [2], Md. Mahabub Hassan [3]
Farazul Haque Bhuiyan[4] , Md Khalilur Rhaman [5]
School of Computer Science and Engineering, BRAC University. Dhaka, Bangladesh

Abstract: *Hundreds of millions of people in the world are hand impaired in some way, and for many, there is no absolute solution. Operation of computers by physically disabled people; especially with hand impairment was quite impossible till now because use of hands plays a vital role in the use of mouse, touch pad and keyboard. We proposed a new system named as "Eagle i-Bot - An eye-controlled system" which has come with a feasible solution for this scenario. With this system, computers and robots can be controlled by the pair of eyes' movement or iris movement and voice commands control all the mouse events. This system works with image processing system based on Voila-Jones algorithm and modified Ada-boost algorithms along with java robot class and sphinx-4 frameworks. In this paper, this system is described including software and hardware aspects, algorithms that are used and scopes where Eagle i-Bot can be used.*

Keywords: *eye-controlled system, iris detection, micro-controller, real time eye detection, voice recognition*

I. Introduction

Eagle i-Bot can be used with comfort by the disabled people (hand-impaired) not only to navigate computers but also to drive interfaced robots with these computers, only moving their eyes or iris. This system introduces a new approach where no mouse is required, only a camera does the job.

In this paper, we first propose a real-time eye-controlled system. Then we show the integration of eye and iris detection algorithm with the electronics circuit and robot, which can be operated by the movement of the pair of eyes or iris. We implemented a new device using micro-controller, designed a user interface, used our own algorithm along with some existing algorithm and made it work. This is a new approach of implementing such system.

This system works with the combination of two different processes, image processing and voice recognition. Image processing includes face detection, pair of eyes detection, iris detection, keeping track of the position of the pair of eyes on the computer screen. It uses a web cam of high resolution. This camera takes live video as input and the system processes each frame. Then, it checks whether there is a face or not. If it finds face, it detects the eye-pair and the iris, the value of (M,N) - the most upper-left corner of the detected rectangle of eye-pair of that frame is sent as the input in mouse-control class. That point works for determining the position of the mouse-pointer. Movement of the pair of eyes is generated by the motion of facial movement but iris movement is independent of it. By moving the pair of eyes at any direction (up, down, left or right), one can hover the mouse on the computer desktop. One can also use the Eagle i-Bot's interface by looking at it (by moving iris) or by moving the pair of eyes. This interface is used to communicate with the circuit and hardware part. Speech recognition is used to click and to use four other basic mouse operations. This is a continuous and robust process. Fig 1 shows all the processes at a glance.

II. Research Method

1.1 Eye Detection

To detect the pair of eyes and get the position of the pair from the frame "Voila-Jones" algorithm has been followed based on rapid object detection. This algorithm is used for frontal face detection [3]. From the frontal face the pair of eyes has been detected in the same frame to reduce false detection. Although there are a couple of Haar Cascade Classifiers available in Open CV library for eye detection, we worked mostly with the pair of eyes and frontal face classifier to acquire precision. Voila-Jones algorithm uses a learning algorithm based on Ada-Boost algorithm, which selects a small number of critical visual features from a larger set. Ada-Boost is an algorithm for constructing a "strong" classifier as linear f(x) =t=1Ttht(x) of "simple" "weak" classifiers ht(x). In this system we are using Haar-traning classifier as the training file. Haar-classifier object detection is based on haar-like features algorithms. These algorithms are used for object detection.

Two kinds of eye detection are used in Eagle i-Bot. It can detect the left most eye of the image or the pair of eyes. For detecting the left most eye, it searches for face in a grabbed frame after converting it into gray scale. If faces show up in that very frame, it tries to detect the eyes setting the region of interest (ROI) within the region of the detected face. Haar Cascade Frontal Face Classifier and Eye classifier are used for face and eye detection respectively, which comes along with the OPEN CV library. [1]

Setting the ROI:

```
cvSetImageROI (grabbedImage, cvRect(r.x(),r.y()+(r.height()/5), r.width(),r.height()/3));
r.x()=upper left corner abscissa of the detected face's rectangle
r.y()=upper left corner ordinate of the detected face's rectangle
r.width()= width of the detected face's rectangle
r.width()= width of the detected face's rectangle
```

It reduces the searching time and makes it faster in processing shown at Fig.1. After the detection of the left eye, it places the cursor on the position of the detected rectangle's upper left corner (M,N) (Figure 3.b) using the methodology "cursor movement" which is explained in section 2.3.

It can also detect the pair of eyes together directly without detecting faces using the pair of eyes classifier which also comes with the OpenCV library. In Eagle i-Bot this classifier is used widely rather than the use of the left most eye detection for the user's convenience, which is understood after conducting some experiments. After this detection Eagle i-Bot places the cursor on the position of the detected pair of eyes' upper left corner (M,N) (Figure 3.a) using the methodology "cursor movement". That is how a user can move the cursor anywhere on the screen and click.

1.2 IRIS DETECTION

Iris is detected setting the ROI within the region of detected eyes. The steps are shown in Fig.4. After the conversion of gray scale, Eagle i-Bot filters it with Gaussian blur and then sends it to Canny-edge detection. At the last step, it uses OpenCV's Hough circle detection to detect the iris. It works very well in real time with good precision. [10][11]

It detects the right grid at which the user is looking at using this calculation and procedure:

A = (Width of the rectangle of the detected pair of eyes + value of x of the upper left corner of the same rectangle)/2

B = (Height of the rectangle of the detected pair of eyes + value of y of the upper left corner of the same rectangle)/2

The center of the circle is (P, Q).

If P is smaller than A and Q is smaller than B, cursor will appear in the middle of grid number one using the methodology "cursor movement" and so on.

The table of graphical illustration is followed by the interface shown in TABLE 2.

After detecting the grid Eagle i-Bot places the cursor on the right grid using the cursor movement methodology. It works very good in case of grid 2,4,6 and in very rare case it detects the iris for other grid position and user cannot use it for hovering the cursor outside of these grids and interface.

1.3 CURSOR MOVEMENT AND EVENT LISTENING METHOD

To move the mouse cursor, we used the Java Robot class [9]. This class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. For each frame grabbed via the web cam, the upper left corner value of the detected eye-pair's rectangle or the detected left

eye's rectangle or the mid-point value of the selected grid (in case of iris) is sought and the system moves the mouse pointer to a certain position on the screen passing it as a parameter to the method mouseMove() of Java Robot Class. This means, if the values are (M,N) , then the cursor will show up in the position of (M,N) on the screen.

After bringing the pointer to a position on computer screen, mouse events can be executed by voice commands using voice recognition. All the operations/events of the mouse have been implemented in Eagle i-Bot.

The mouse events have been called with voice commands using a voice recognition system via a microphone. To recognize voice sphinx-4 framework is used. Sphinx-4 is a state-of-the-art speech recognition system written entirely in the programming language. It was created via a joint collaboration between the Sphinx group at Carnegie Mellon University, Sun Micro-systems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with invaluable contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT). [6][7][8]

As mentioned above, voice commands are implemented for mouse-events with the help of the methods of Java Robot class and speech recognition (Sphinx 4). So the five basic operations of mouse can be done by giving voice commands using the microphone, we programmed the voice commands as follows:

left click = single click ;
right click = right click ;
double click = open click ;
grab = grab;

release = release;

To elaborate, after bringing the mouse-pointer on an icon, if anyone says "single click", that particular icon will get selected. Other operations will be done in the same way. One can drag, release, double click or right click anywhere using voice commands.

1.4 Circuits and Mechanical Entity

A RC controlled car is interfaced with Eagle i-Bot which can be controlled by switching the RC circuit (Figure 7). The heart of Eagle i-Bot circuit is a micro-controller. It uses a micro-controller of Microchip brand (PIC16F877A). A crystal is required to drive the micro-controller. In Eagle i-Bot a crystal of 8 MHz and some NMOS's (IRF3205) are used for switching purpose. The RC circuit of the car is used via switching, resulting the wireless car to turn left, right, backward, forward, left forward, right forward, left backward, right backward and off. When any button is clicked of the Eagle i-Bot's interface (Figure 4), it sends the signal to the micro-controller via RS-232 protocol and it does switching operations according to that command setting the output high and low of the micro-controller. When the output pin of the MCU is high, it shorts the drain and source of NMOS and drives the car. When all the pins are low, the car is turned off. This table shows in TABLE 1 the pin configuration we used in Eagle i-Bot.

III. Results And Analysis

Eagle i-Bot performs very well in real time continuously and flawlessly [12]. This system has been experimented with five hand impaired people and five ordinary people in a couple of indoor environments. It ran better in real time with the movement of the pair of eyes which is detected directly without performing face detection first. The algorithm of iris detection of this system was independent of environment, lighting condition and iris color and worked better than the threshold algorithm ([2],[4],[5]) to some extent. The interface worked very well in case of grid 2, 4, and 6 and in very rare case it detected the iris for other grid position. It provided a wholesome usability of computers and robots for hand impaired people.

IV. Figures And Tables

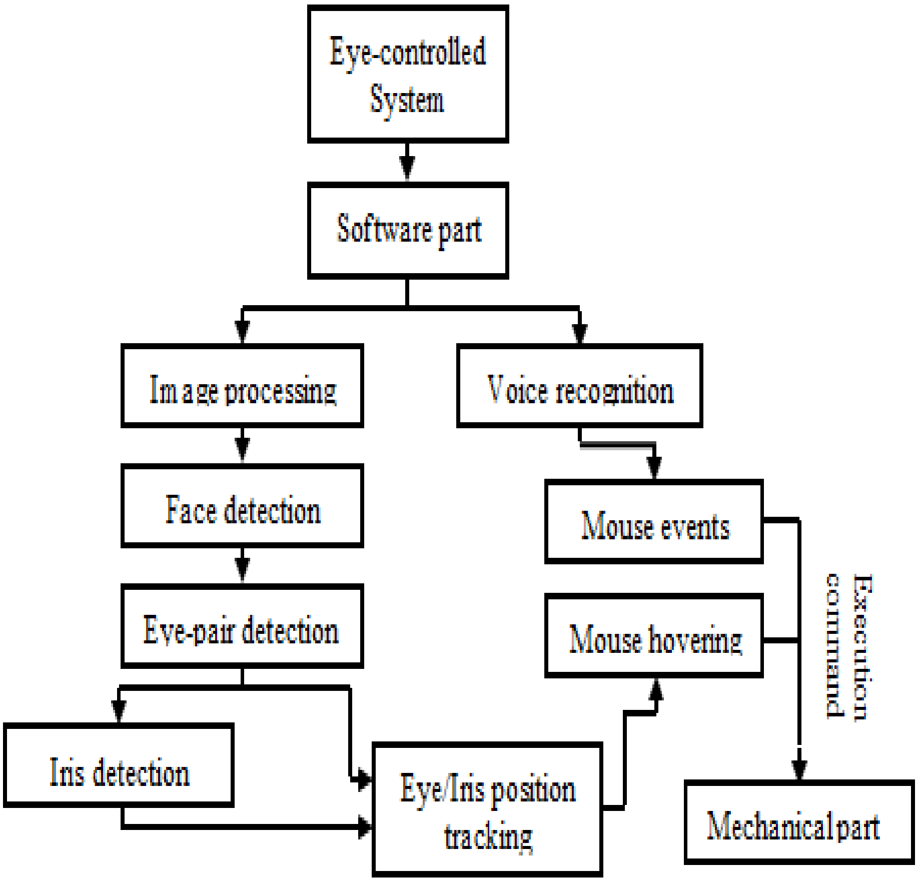


Fig. 1: Eye-controlled system work-flow

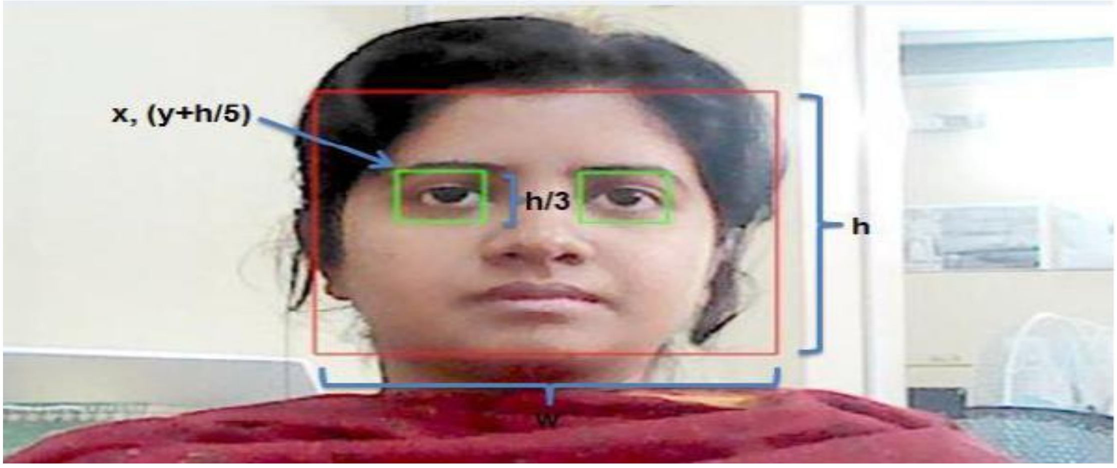


Fig. 2: eye detection

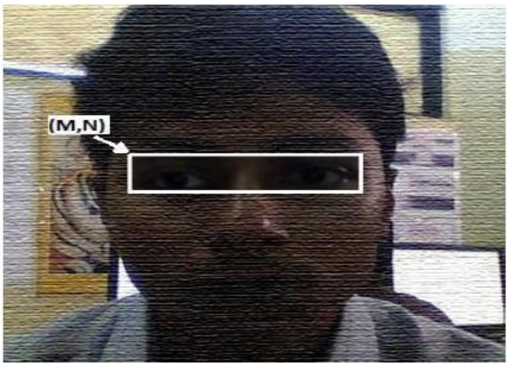


Fig. 3.a: eye-pair detection

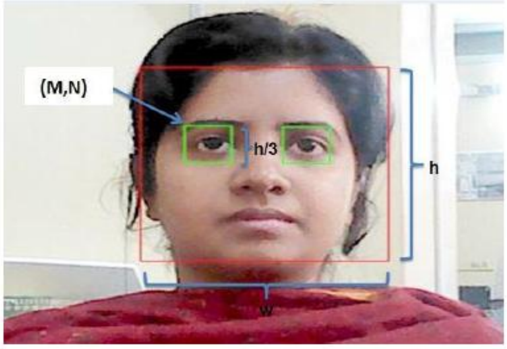


Figure 3.b: leftmost eye detection

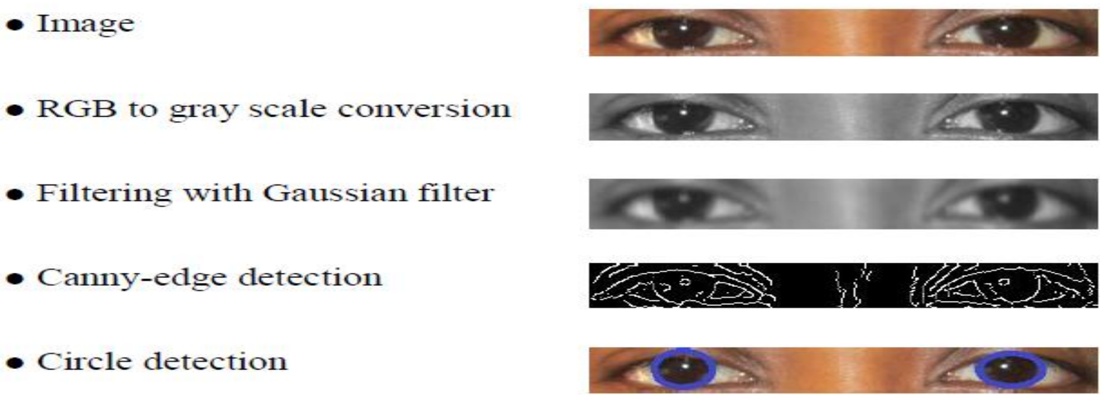


Fig. 4: steps of Iris detect setting the ROI within the region of detected eyes

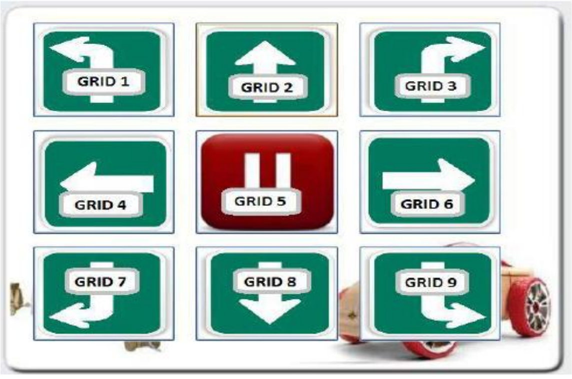


Fig. 4.a: interface of Eagle i-Bot

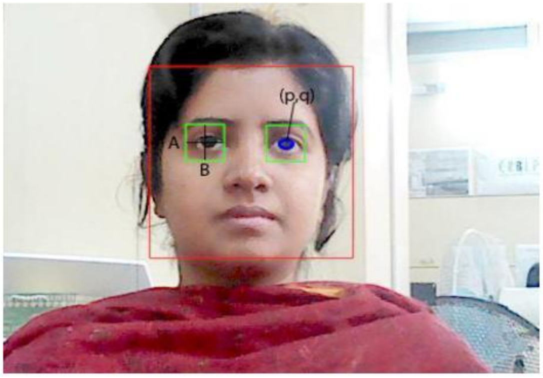


Fig. 5: Grid Selection

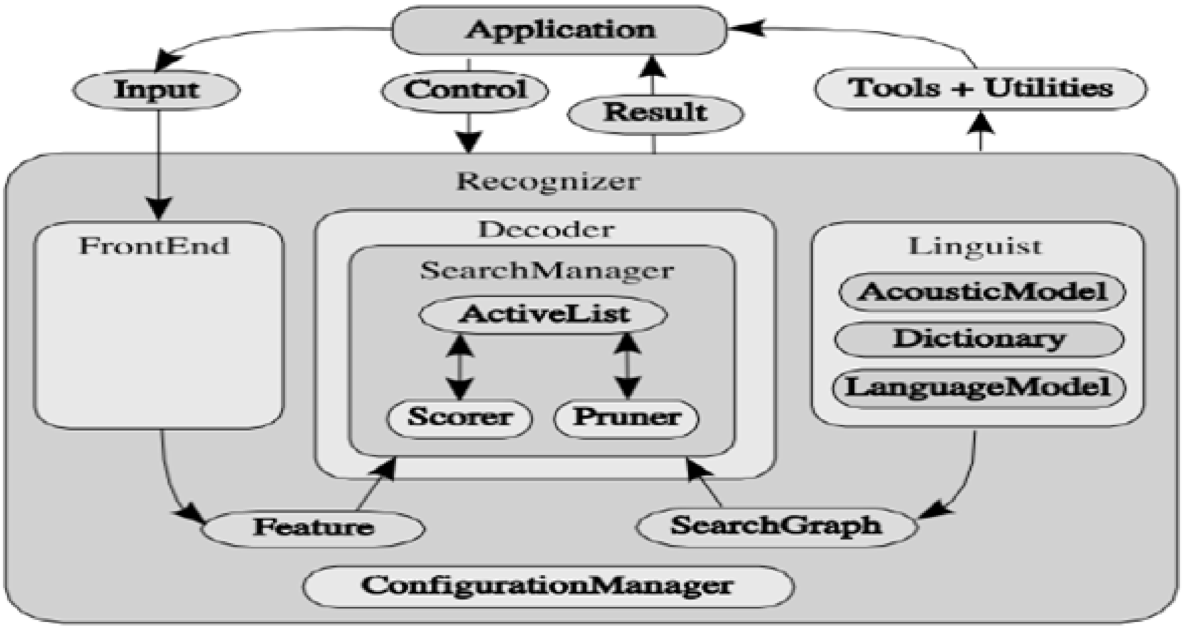


Fig. 6: sphinx-4 decoder framework [6] [7] [8]

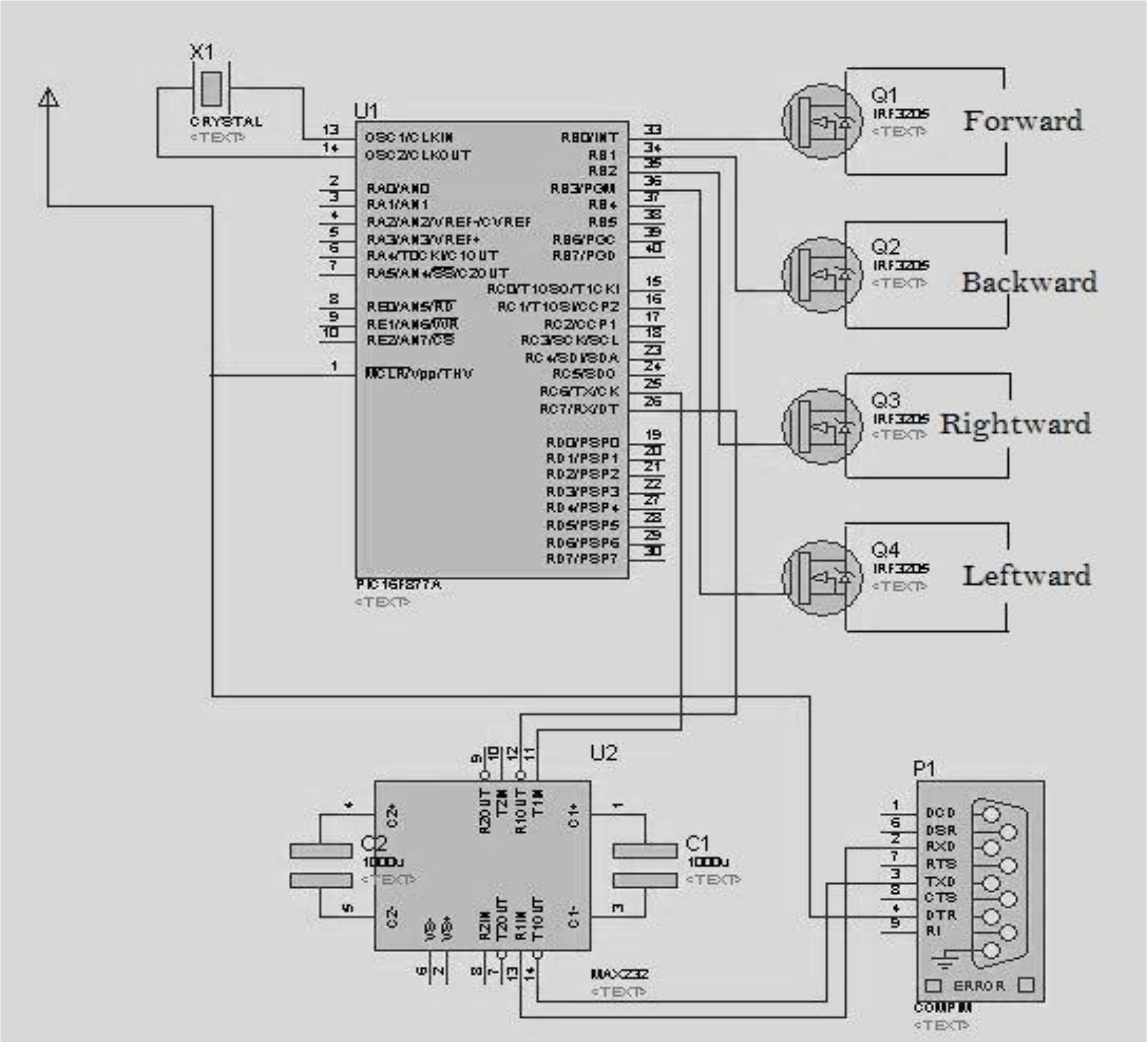
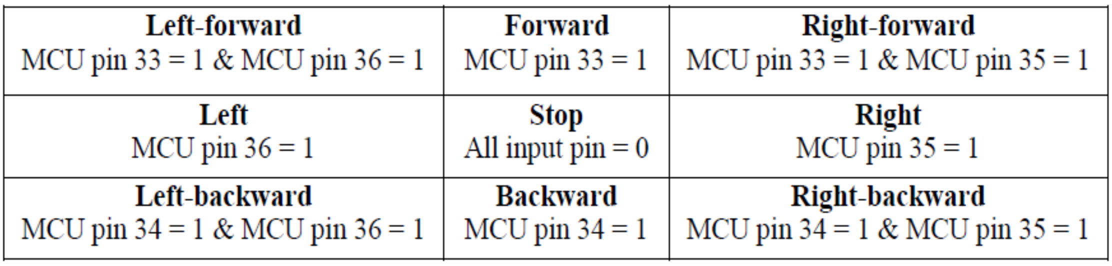


Fig. 7: Switching circuit

TABLE 1: Micro-controller inputs



Left-forward MCU pin 33 = 1 & MCU pin 36 = 1	**Forward** MCU pin 33 = 1	**Right-forward** MCU pin 33 = 1 & MCU pin 35 = 1
Left MCU pin 36 = 1	**Stop** All input pin = 0	**Right** MCU pin 35 = 1
Left-backward MCU pin 34 = 1 & MCU pin 36 = 1	**Backward** MCU pin 34 = 1	**Right-backward** MCU pin 34 = 1 & MCU pin 35 = 1

TABLE 2: The table of graphical illustration

GRID1 P<A & Q<B	**GRID2** P=A & Q<B	**GRID3** P>A & Q <B
GRID4 P<A &Q=B	**GRID5** P=A & Q=B	**GRID6** P>A & Q =B
GRID7 P<A &Q>B	**GRID8** P=A & Q>B	**GRID9** P>A & Q>B

V. Conclusion

This is a very feasible solution for the hand-impaired people because it requires only a web cam and microphone as main input devices and they come handy. The circuit can be customized to drive a wheelchair which will serve the purpose of movement of hand and limb impaired people. This circuit or any modified version of this can be used to turn on and off electronics devices or to drive any kind of mechanical entity.

References

Web-sources:

[1] Nashruddin, *OpenCV Eye Detection*, http://nashruddin.com/OpenCV_Eye_Detection , Last visited July 2012

[2] Face Detection using OpenCV, http://opencv.willowgarage.com/wiki/FaceDetection , Last visited July 2012

[3] Sphinx-4, http://cmusphinx.sourceforge.net/sphinx4/javadoc/index.html?overview-summary.html

[4] Sphinx-4, A speech recognizer written entirely in the JavaTM programming language, http://cmusphinx.sourceforge.net/sphinx4/

[5] Puneet Kalra, *Speech Recognizer In Java*, http://puneetk.com/speech-recognizer-in-java-tutorial

[6] java.awt - Class Robot, http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/Robot.html

[7] Iris Detection, http://codingbox.net/jinstall/index.php?option=com_content&view=category&id=11&layout=blog&Itemid=22

Journals:

[8] J. Daugman. *How iris recognition works*. IEEE Transactions on circuits and systems for video technology, 14(1):21{30, 2004}

[9] N. Otsu. A *threshold selection method from gray-level histograms*. Automatica, 11:285{296, 1975}

Theses:

[10] Boyd, Carmaciu, Giannaros, Payne, Snell, Gillies, *Iris Recognition*, Imperial College London, *March 19, 2010*.

Demonstrations:

[11] Iris detection demonstration, http://youtu.be/9tfabCRYj-4

[12] Eagle i-bot, http://youtu.be/3cZXdjH9fbc