

Prototyping An Autonomous Eye-Controlled System (AECS) using Raspberry-Pi on Wheelchairs

Jayanth Thota

University of Houston Clear Lake
2700 Bay Area Blvd
Houston, TX 77058

Priyanka Vangali

University of Houston Clear Lake
2700 Bay Area Blvd
Houston, TX 77058

Xiaokun Yang

University of Houston Clear Lake
2700 Bay Area Blvd
Houston, TX 77058

ABSTRACT

In order to help physically disabled persons to make their life independent, this paper proposes an autonomous eye controlled system (AECS) on wheelchairs. In this work, several OpenCV image processing algorithms are employed to track the eye motion to coordinate the wheelchair moving left, right, and straight forward. We use the Raspberry-Pi B+ board as the system center to process the images and control the motors via GPIO. Experimental results show that the ACES system can be effectively used in the prototype, and outperforms the hand gesture controlled system by 25% processing latency reduction.

General Terms

Computer engineering, embedded system design

Keywords

Eye controlled system, gesture controlled system, image processing, OpenCV, Raspberry-Pi

1. INTRODUCTION

Today, autonomous wheelchairs are enhancing physical abilities of the elderly and disabled and making their life more accessible. This paper presents a novel implementation of an autonomous system for the completely disabled persons, which allows them to control wheelchairs by eye movement.

As the system structure shown in Figure 1, a camera is mounted in front to capture the image of any one of the eyes (either left or right) and track the position of eye pupils with the use of many image processing algorithms. According to the position of the eye, wheel chair motors are directed to move left, right, and forward.

The main part of the hardware system is the Raspberry-Pi board, which is utilized to perform the image processing and control the hardware system. It captures the frames of images in real time and extracts the commands from the eye motions. Then, the Raspberry-Pi sends the control signals to engines to perform the specific operation, such as running the motors in clockwise direction, anti-clockwise, and stop the motors. On the wheelchair, an ultrasonic sensor for obstacle detection is additionally implanted for safety. It can notify the Raspberry-Pi and help to stop the motors in the event that an obstacle near the wheelchair is detected.

The most challenge part of this system is the eye-motion detection algorithm. An open computer vision (OpenCV) library with numerous computer vision algorithms is used for the face and eye recognition [11]. For example, the component location calculation is used to identify single or numerous faces and both eyes. To find the correct eye pupil and its inside point is extreme objective of this algorithm. A few applications and calculations are further employed to discover the precise pupil area, and several sensor based recognition strategies, such as ECG, EEG, and EOG, are utilized to consequently discover eye understudy and follow eye pupils. The main contributions of this system are:

- We propose a software-hardware cooperative system for the eye-motion controlled wheelchair, in order to help physically handicapped people to make their life independent.
- Several image and video processing algorithms in view of face, eye, and eye pupil motion identification with least deferral of time are presented. Prototyping results show that the AECS system reduces the processing latency (less than 3 seconds) to 75% compared with the hand gesture controlled system proposed in [10] (4 seconds).
- Safety issue is further considered with the highest priority in this work to avoid the occasion of any collision. An ultrasonic sensor is implanted to detect the obstacles and stop the motion of the wheelchair automatically.

The rest of the article is structured as follows: Section 2 presents related work and Section 3 discusses our proposed system architecture involving both hardware and software. In Section 4 we present the image processing algorithms utilized in this work. We demonstrate the system and evaluate the approach with simulation data in Section 5. Last, we summarize our work in Section 6.

2. PREVIOUS WORK

Currently, there are many control systems creating specific applications for individuals with different issues and incapacitates, such as the hand gesture controlled system [10], infrared light [15], ultrasonic and body kinematics [3], and so forth [7, 4]. The drawbacks in these systems are: 1) they cannot be used by people of higher disability because they require accurate controls; 2) the infrared light based method [15] gives exact identification of the eye understudy focus area and additionally tracks the eye development. The infrared radiations, however, may influence the users' perceivability.

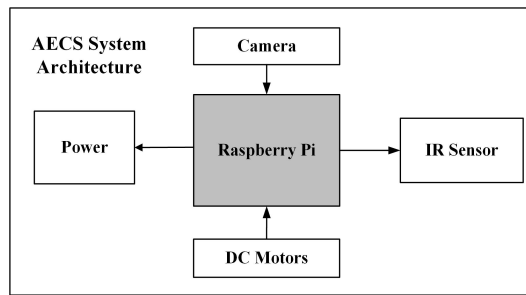


Fig. 1. System Structure.

Moreover, systems based on voice recognition have been widely used today [13, 2, 5]. The main objective of these systems is that move the wheelchair in particular direction based on the voice command. However, in some noisy conditions these systems may fail due to the difficulty for user to convey voice message to the controllers.

To overcome the aforesaid issues, the eye control system gives the freedom to make their life simple and helpful. It detects eye pupils' movement by using some motion sensors, such as ECG, EEG, and EOG [1], and a set of image processing arithmetics to control the wheelchair. First of all, the wavelength of the white portion is recorded by the eyeball sensor. Then, when the user need to move right side, his left eye demonstrates no variety in wavelength however in the left eye the dark bit is detected by the sensor, which prompts diminish in the wavelength. Similar component happens in the right eye as well.

In what follows, many image processing algorithms from the OpenCV library [11] are utilized for face and eye motion discovery. For instance, the haar cascade arithmetic [12] is used to detect single or multiple faces and both eyes. To automatically find out and track the eye pupils' movement, the object detection [14], edge detection [9], and pattern recognition [8] are further employed in this study.

3. SYSTEM ARCHITECTURE

3.1 Hardware Structure

Figure 2 shows the hardware architecture with all the main components, including power supply, Raspberry-Pi board, cameras, motors, and sensors. We also have a Wi-Fi module which helps in connecting the Raspberry-Pi to Internet and communicating using Internet in some emergency situations.

The power circuit gives the possible power supply to individual segments, involving the Raspberry-Pi, camera, sensor, and engines. Second, the system is implemented based on real-time data acquisition. A low power consumption Raspberry-Pi B+ board, as shown in Figure 3(a), is used as the control center of the whole system. It provides input & output pins, USB, UART, HDMI ports and Ethernet adapter port for connecting it through Internet via wired or wireless connection using Wi-Fi adapter [18].

The Raspberry-Pi has a Broadcom BCM2835 framework on a chip, which incorporates an ARM1176JZF-S 700 MHz processor with a video core GPU, and is initially transported with 512 MB of RAM and capable of up to 32 GB external memory. It works very efficiently with multiple images provided by the web camera connected with the Raspberry-Pi board. Distance between eye and the camera is 10 to 14 cm. The camera interfaces to the board using the UV4L driver. In this work, not only the applications, but also the

system performance is considered [16, 17]. Hence, a 720P video format camera is used as a balance among the memory cost, processing speed, and recognition accuracy.

Figure 3(b) shows our prototyping demo of the auto-controlled wheelchair. The motor driving circuit, L293D, is connected with the Raspberry-Pi, the power supply of motors, and the relay for controlling the motor driving integrated circuit. The Raspberry-Pi center continuously generates command signals to enable the GPIO pins and performs the operations of left, right, forward, and stop. Furthermore, safety issues is highlighted in this work. As shown in the Figure, there is an infrared sensor mounted in front of the wheelchair to recognize the obstacles and stop the wheelchair with the highest control priority.

3.2 Software Design

The Raspberry-Pi has its own operating system – Raspbian. It runs python programs on Linux where it can be associated with primary board. The Win32 disk imager software is used in this work to boot a Raspberry image file. While putting a bootable memory device on Raspberry-Pi board, it can access the Raspbian directly.

The main part of the image processing algorithms is completed with the assistance of the OpenCV 3.0.0 library [11]. OpenCV is released under a BSD license and it is thus free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS, and Android. It is designed for computational efficiency and with a strong focus on the real-time applications. Enabled with OpenCV, our proposed work can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

3.3 Operational Flow Chart

The AECS system begins with catching pictures consistently from a web camera. By analyzing the captured face and eye images, the system crops the eye region of interest and identifies all conceivable circle displayed on the specific area. In what follows, the separation between the center point and eye circle center point is measured using the coordinate system logic. A base separation shows the eye pupil introduced in left and most extreme qualities demonstrates the eye proceeded onward right.

Figure 4 shows the operational flow chart. It is organized into blocks having an internal structure formed from the three fundamental elements: a state box, a decision box, and a conditional box. State boxes are rectangles, conditional boxes are rectangles with round corners, and decision boxes are diamond-shaped.

The state box represents the status of the system, involving Initial state, Obstacle Detection state, and Move Detection state in this flow chart. The values in the decision boxes determine the possible

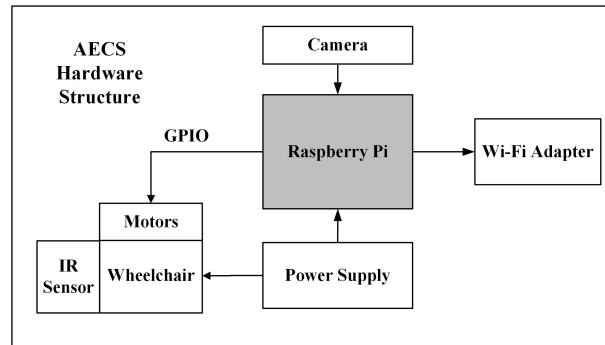
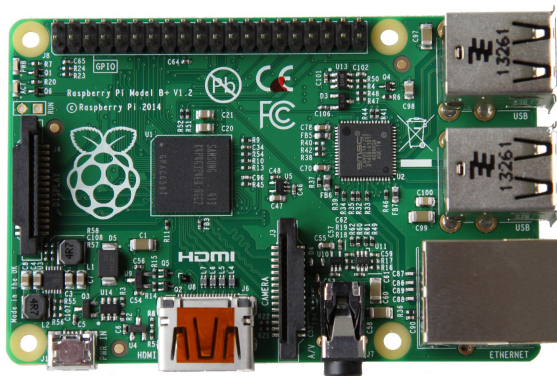
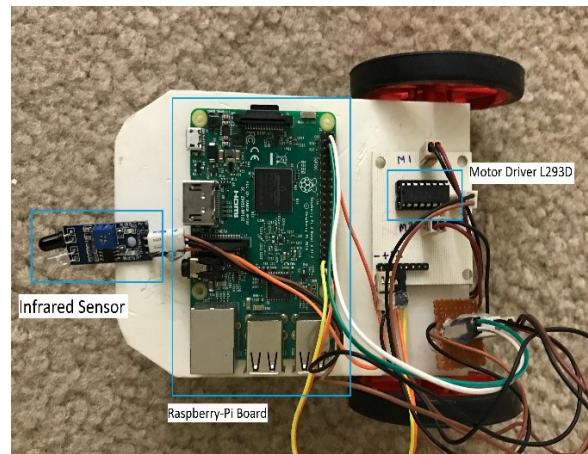


Fig. 2. Hardware System Architecture.



(a) Raspberry-Pi B+ Board



(b) Hardware Demo

Fig. 3. Hardware System

paths through the block under the action of the inputs. For example, if there is no developments of the eye, then it goes back to the Initial state. In the state of Obstacle Detection, the wheelchair must be immediately stopped in the event that an obstacle is detected near to the wheelchair, since the safety issues has the highest priority in this system design. If there is no obstacle detected, the flow chart goes into the Move Detection state. In this state, both engine move when the eye is in center, and the wheelchair moves forward bearing. Likewise, when eye development is left, a wheelchair left side engine runs; when eye recognition is right, the right motor runs.

For the begin and stop operations, eye flickering to control the wheelchair is used. In the case that the eye shut for 3 seconds, the system totally stops and at the end of the day it will close the eye for 3 seconds, the system is then reactivated.

4. IMAGE PROCESSING ALGORITHMS

To distinguish the correct eye pupil area is exceptionally challenging in this work. In this study, many algorithms are utilized for various applications, such as the haar cascade, hough change, and edge recognition. The main program is shown in Algorithm 1. Initially, all the control pins, MOTOR_1, MOTOR_2, and obstacle pins, need to be assigned. Then, the system needs to extract the frames of im-

ages once videos are captured, detect the face and eye location, and mark the eye region using the haar cascade arithmetic. If there are multiple faces or no frames in the window, it falsely detects the face and sends an "face not detected" command to the processor.

Algorithm 1 Main Program

```

1: IR Obstacle PIN Assignment;
2: MOTOR_1 PIN Assignment;
3: MOTOR_2 PIN Assignment;
4: Face_Cascade.Load( face_cascade_name );
5: Eyes_Cascade.Load( eyes_cascade_name );
6: cv::namedWindow(face_window_name,CV_WINDOW_NORMAL);

7: CvCapture * capture = cvCaptureFromCAM();
8: if FrameCapture then
9:   while TRUE do
10:    Detect and Display Frame
11:   end while
12: else
13:   Break
14: end if
  
```

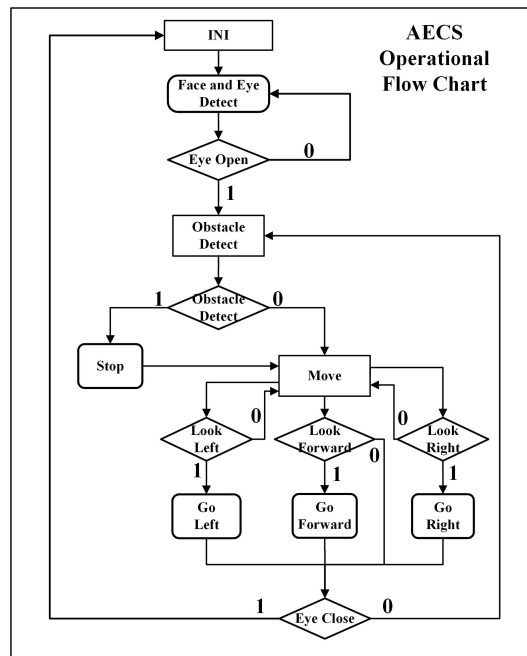


Fig. 4. Hardware Operational Flow Chart.

After detecting the eye, it creates a box along with the eye and determines the center point of the eye. The next step is detecting the eye pupil and analyzing the command given by the images. It decides in which direction the wheelchair should move. Figure 5 represents the algorithm flow chart based on the prototyping. More specifically, they are briefly introduced below.

4.1 Frame Extraction and Background Subtraction

The extraction of valid information from the video is applied in order to process video data efficiently. Here, the system also employs the background subtraction technique to reduce the transfer stress of the hardware.

4.2 Image Down-Scaling

Before put into the system, all the colored images are converted to gray-scale format in this work. In such a way, the amount of pixels processed by the system is further reduced to 33% compared with the raw RGB images, so as to improve the response speed of the whole system.

4.3 Face and Eye Detection

In this study, the haar course calculation is utilized for the face and eye identification [12]. After recognizing the faces, it attempts to identify the eye inside the face area and draws the rectangular box over the eyes.

4.4 Image Blurring and Feature Detection

To detect the exact edges of eye pupils, the system apply the gaussian blur filter to blur the image and further reduce the processing latency [6]. In Figure 6(a), the first picture shows the gaussian blur filtered image and the second shows the features detected based on the image.

4.5 Edge Detection and Eye Pupil Circling

Consequently, the system use the canny edge detection and corner edge recognition algorithms [9] to decide the edges of the eye pupils, which is shown in Figure 6(b). In what follows, a circle is then drew on the eye pupils using the hough circle transform arithmetic [19]. It helps to identify the eye pupils with the images input to the controller.

4.6 Eye Motion Tracking

A coordinate system is finally applied to track the eye movements [14]. As shown in Figure 7, the horizontal axis denoted as X and vertical axis denoted as Y, respectively, represent the eye movements in left or right directions. Assume that A0 and A2 represent the corner points of the eye pupil in the X direction, and B0 and B2 indicate the corner of the eye pupil in the Y direction. Therefore, the eye pupil position, denoted as coordinate point (A1, B1), can be calculated as

$$A1 = (A0 + A2)/2 \quad (1)$$

and

$$B1 = (B0 + B2)/2. \quad (2)$$

Algorithm 2 shows the basic eye pupil motion detection program, by which the system works according to the position of the eye pupil and performs wheelchair movement in left, right, and forward directions.

5. EXPERIMENTAL RESULTS

This section mainly shows the prototyping and experimental results. First of all, by analyzing the motion of the eye pupils, the Raspberry-Pi controller can successfully send out signals to the

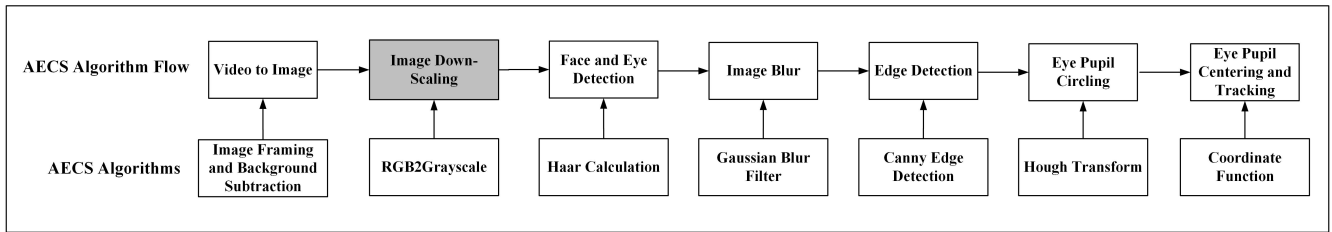


Fig. 5. Image Processing Flow Chart.

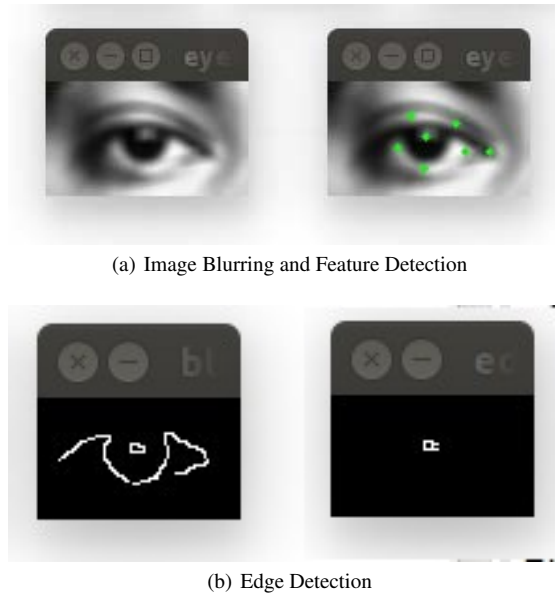


Fig. 6. Algorithms

Algorithm 2 Detect and Display Frame Program

```

1: LeftRightCornerRegion.width -= Pupil.x;
2: LeftRightCornerRegion.x += Pupil.x;
3: LeftRightCornerRegion.height /= 2;
4: LeftRightCornerRegion.y += LeftRightCornerRegion.height / 2;
5: if ObatacleDetect then
6:   MOTOR_1 OFF; MOTOR_2 OFF;
7:   Printf STOP;
8: else if LeftRightCornerRegion.x <0 then
9:   MOTOR_1 LOW; MOTOR_2 HIGH;
10:  Printf RIGHT TURN;
11: else if LeftRightCornerRegion.x >0 then
12:  MOTOR_1 HIGH; MOTOR_2 LOW;
13:  Printf LEFT TURN;
14: else if LeftRightCornerRegion.x = 0 then
15:  MOTOR_1 HIGH; MOTOR_2 HIGH;
16:  Printf FORWARD;
17: end if
  
```

motor driver circuit to perform the motion of the wheelchair in the desired directions. Meanwhile, if any obstacle is detected by the infrared sensor, the system can stop the wheelchair immediately by sending out a stop signal to motors. The obstacle detection has the

highest priority, so the engines are stopped no matter which states the engine is inside. More specifically, the function of the entire system is verified by the following parts:

- capturing the images via the camera,
- determining location of the eye pupils of both eyes,
- using that data to perform wheelchair motion,
- stopping the system on detection of any obstacle.

All these functions are shown in Figure 8. First, Figure 8(a) demonstrates the forward moving test. Likewise, turning left and right motions based on the eye-control system are shown in Figure 8(c) and Figure 8(d), respectively. As aforementioned, the system uses the ultrasonic sensor to recognize obstacles and effectively measures the distance between the wheelchair and obstacles. The result that motors are stopped due to an obstacle detection is demonstrated in Figure 8(b). Finally, Figure 8(e) shows the test vector that the user is not facing the camera.

Furthermore, the processing speed from the eye movement to the engine control is tested. It can be observed that the average latency is less than 3 seconds for the entire operations. And in the worst case, the command shown on the screen has a 3-second delay after sending the commands by eye movement.

In sum, experimental results show that our proposed eye-controlled system can be effectively used in this prototyping wheelchair.

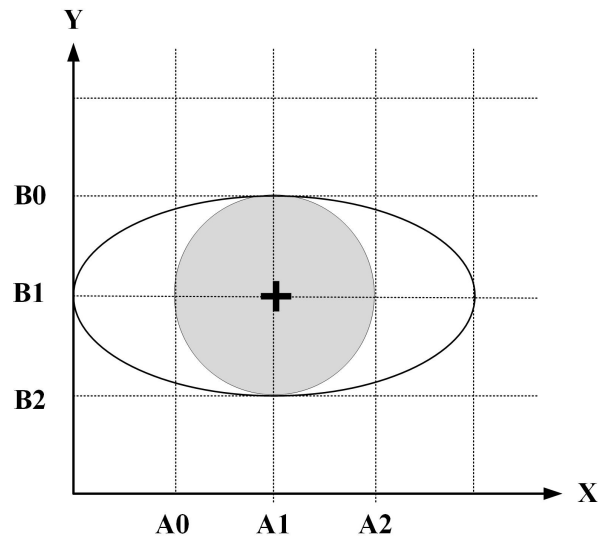


Fig. 7. Eye Motion Tracking

Moreover, the AECS system reduces the processing time to 75% compared with the other hand gesture controlled systems such as [10] (4 seconds).

6. CONCLUSION

This paper proposes a software-hardware co-design system on wheelchairs to reduce the need of help required by the handicapped individuals. Several digital image processing algorithms from the OpenCV library are employed in this work, such as background subtraction, RGB to grayscale conversion, haar cascade, hough change, gaussian blur filter, and edge recognition. Experimental results show that our proposed AECS system can be effectively used on the wheelchair prototype and the optimized algorithm reduces the execution time to 75% compared with the traditional methodologies.

However, the system performs the wheelchair movement operation with 3-second delay time. It is not acceptable as an commercial usage. In addition, to track the eye pupil in the low-light condition is another big challenge for the AECS system. That is one reason that the other auto-controlled wheelchairs usually applied very complex and combined systems with multiple technologies to improve the detecting accuracy, although sometimes it is a conflict with the system speed and resource cost. In the future, therefore, instead of processing algorithms with Raspbian, FPGA-based implementations such as RGB2Grayscale and Edge Detection will be employed on gate-level chips in order to improve the speed.

7. REFERENCES

- [1] A. Ahamed, A.U. Ahad, H.A. Sohag, and M. Ahmad. Development of low cost wireless biosignal acquisition system for eeg, emg and eog. *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT2015)*, December 2015.
- [2] C. Aruna, A. Dhivya Parameswari, M. Malini, and G. Gopu. Voice recognition and touch screen control based wheel chair for paraplegic persons. *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCC2014)*, March 2014.
- [3] P. Britto, J. Indumathi, S. Sivarasu, and L. Mathew. Automation of wheelchair using ultrasonic and body kinematics. *National Conference on Computational Instrumentation (NCCI2010)*, pages 19–20, March 2010.
- [4] A. Dev, H.C. Chacko, and R. Varghese. Eye controlled wheel chair using eog. *International Conference on Computing and Control Engineering (ICCC2012)*, April 2012.
- [5] M.A. Devi, R. Sharmila, and V. Saranya. Hybrid brain computer interface in wheelchair using voice recognition sensors. *2014 International Conference on Computer Communication and Informatics (ICCCI2014)*, January 2014.
- [6] E.S. Gedraite and M. Hadad. Investigation on the effect of a gaussian blur in image filtering and segmentation. *53rd International Symposium ELMAR-2011*, pages 393 – 396, September 2011.
- [7] S.U. Kumar and V.M. Vinod. Eog based wheelchair control for quadriplegics. *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS2015)*, March 2015.
- [8] Y. Li, X. Xu, N. Mu, and L. Chen. Eye-gaze tracking system by haar cascade classifier. *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA2016)*, June 2016.
- [9] P. Melin, C.I. Gonzalez, J.R. Castro, O. Mendoza, and O. Castillo. Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 22(6):1515–1525, January 2014.
- [10] S.D. Mhaske and M.P. Dale. Hand gesture controlled wheelchair system using image processing. *International Journal of Electrical, Electronics and Computer Systems*, 4(2):2347–2820, February 2016.
- [11] OpenCV. Open source computer vision. <http://opencv.org/>.
- [12] M. Rezaei, H.Z. Nafchi, and S. Morales. Global haar-like features: A new extension of classic haar features for efficient



Fig. 8. Eye Controlled Wheelchair Results

- face detection in noisy images. *6th Pacific-Rim Symposium on Image and Video Technology (PSIVT 2013)*, 8333:302–313, 2013.
- [13] M.F. Ruzajj, S. Neubert, N. Stoll, and K. Thurow. Design and testing of low cost three-modes of operation voice controller for wheelchairs and rehabilitation robotics. *2015 IEEE 9th International Symposium on Intelligent Signal Processing Proceedings (WISP2015)*, May 2015.
- [14] K.S. Sabarish and A.M. Suman. Eyeball and blink controlled robot with fuzzy logic based obstacle avoidance system for disabled. *International Conference on Electrical and Electronics Engineering (ICEEE2012)*, pages 16 – 17, June 2012.
- [15] Y.G. Shin, K.A. Choi, S.T. Kim, and S.J. Ko. A novel single ir light based gaze estimation method using virtual glints. *IEEE Transactions on Consumer Electronics*, 61(2):254–260, July 2015.
- [16] X. Yang and J. Andrian. A low-cost and high-performance embedded system architecture and an evaluation methodology. *2014 IEEE Computer Society Annual Symposium on VLSI (ISVLSI2014)*, pages 240–243, March 2014.
- [17] X. Yang and J. Andrian. A novel bus transfer mode: Block transfer and a performance evaluation methodology. *Integration, the VLSI Journal*, 52(C):23–33, January 2016.
- [18] X. Yang, X. Niu, J. Fan, and C. Choi. Mixed-signal system-on-chip (soc) verification based on system verilog model. *The 45th Southeastern Symposium on System Theory (SSST 2013)*, pages 17–21, March 2013 2013.
- [19] Bing Z. Using vector quantization of hough transform for circle detection. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA2015)*, pages 393 – 396, December 2015.