Computer-Vision Algorithms to Steer Powered Wheelchairs

M Haddad1, D Sanders1, G Tewkesbury1, M Langner2, S Muhyaddin3 and M Ibrahim4

1 School of Mechanical and Design Engineering, University of Portsmouth. Anglesea Building, Anglesea Road, Portsmouth PO1 3DJ, UK

2 Chailey Heritage Foundation. North Chailey, Lewes BN8 4EF, UK

3 North Wales Business School, Wrexham Glyndwr University. Wrexham LL11 2AW, UK

4 Ministry of Communications. Abo-Naws Street, Bagdad, Iraq

Email: [malik.haddad@port.ac.uk](mailto:malik.haddad@port.ac.uk)

**Abstract.** A novel approach to steer powered wheelchairs using Computer-Vision algorithms, a camera module and a Raspberry Pi is presented. When a powered wheelchair user made a movement, the new approach detects that movement and uses it to operate a powered wheelchair. A circuit is created connecting a camera module, relay and a Raspberry Pi. Python programming language is used to create a program that detects movement of a powered wheelchair user. Three different Computer-Vision algorithms are considered: Background-subtraction, Python-Imaging-Library and Open-source-Computer-Vision (OpenCV). The camera was directed towards user a body-part used for controlling a wheelchair. The new approach will detect and analyse movement of the body-part and activate a wheelchair motor accordingly. Two User Interfaces were created: a simple User Interface to operate the approach and a technical interface to control the sensitivity, frame rate, size of image captured and detection area, and duration of wheelchair motor activation. Practical testing showed that OpenCV provided the highest accuracy and sensitivity among the three algorithms. The new approach successfully detected users’ voluntary movements and translated those movements into driving-commands used to drive a wheelchair. Future clinical-trials will be conducted at Chailey Heritage Foundation.

2. Introduction

A novel approach for operating a powered wheelchair using an image processing algorithm and a Raspberry Pi is presented. The work presented is part of research conducted by the authors at the University of Portsmouth and Chailey Heritage Foundation funded by the Engineering and Physical Sciences Council (EPSRC) [1]. The main aim of the research is to apply AI techniques to powered mobility to improve the quality of life of powered mobility users.

Around 15% of the world population have been experiencing some type of disability with 2 to 4% of them diagnosed with major problems in mobility [2]. Population ageing and the spread of chronic disease have increased these numbers [2,3]. The type of disability is shifting from mostly physical to a more complex mix of physical/cognitive disability. New systems, transducers and controllers are required to address that shift. People with impairment often had lower quality of life than others [4]. New Smart input devices are required that use dynamic movement of body-parts using new contactless transducers to consider users’ level of functionality rather than the type of disability and determine user intentions.

George Klein created the first powered wheelchair in collaboration with the National Research Council of Canada to help quadriplegics wounded during the Second World War [5]. Since then, powered mobility often became a preferred option for people with disability [5]. Many researchers worked on improving navigation and steering of powered mobility by creating new systems. Sanders et al. [6,7] used a sensor system to control wheelchair-veer and enhance driving. Many researchers used zero forced sensing switches to operate powered wheelchairs [8]. Sanders and Bausch [9] used an expert-system that analysed users’ hand tremor to improve driving. Sanders [10] considered self-reliance factors to develop a system that shared control between powered wheelchair drivers and an intelligent ultrasonic sensor system. Haddad et al. [11-13] used readings from ultrasonic sensor arrays as inputs to a Multiple Criteria Decision Making (MCDM) system and blended the outcome from the MCDM system with desired input from a user to deliver a collision-free steering direction for a powered wheelchair. Haddad and Sanders [14] used a MCDM method, PROMETHEE II, to propose a safe path. Haddad et al. [15,16] utilized microcomputers to develop intelligent Human Machine Interface (HMI) that safely steered powered wheelchairs. Many researchers created intelligent systems to collect drivers’ data for analysis [17-21] and applied AI techniques to powered mobility problems, deep learning to safely steer a powered wheelchair [2], rule-based systems to provide a safe route for powered wheelchairs [22], intelligent control and Human Computer Interfaces based on expert systems and ultrasonic sensors [23], image processing algorithms and facial recognition to identify powered mobility drivers [24,25]. The system created aimed at improving the quality of life of powered wheelchair users and enhancing their mobility.

Powered wheelchair users often produced a voluntary movement to activate an input device used to drive their wheelchair. Users often used a joystick or a switch to indicated the desired direction and speed. Examples of other input tools included foot-control, head/chin controllers or sip-tubes and lever switches.

Interviews with Occupational Therapists (OTs), carers and helpers at Chailey Heritage Foundation/School showed that some students lacked the ability to provide enough hand movement to use a joystick or a switch and the clicking noise produced from closing switches disturbed the attention of some of the young wheelchair drivers diagnosed with cognitive and physical disability.

The approach presented in this paper considered a new way to operate a powered wheelchair using a minimal amount of limb/thumb movement and zero force sensing. The new approach aimed to detect users’ hand/thumb movement and use that movement to operate a powered wheelchair.

1. The New Approach

The new approach used an electronic circuit and a Python program to detect movement and operate a powered wheelchair. The circuit connected a camera with a Raspberry Pi and a relay. The camera was directed to user body part responsible to generate a voluntary movement.

The program was installed onto the Raspberry Pi and triggered the camera to continuously take images. Three Computer-Vision algorithms were considered. A simple UI with four buttons was created to operate the new system as shown in figure 1. The simple UI was designed to match the level of functionality of potential users who had an intellectual disability. It had a straightforward operation and offered a appropriate match between desired commands and user abilities [26].

To detect movement, the Start button shown in figure 1 would be pressed. Due to safety concerns, the system would not start detecting movement until the Start button was pressed.

|  |  |
| --- | --- |
|  | **Figure 1.** UI used to operate new approach. |

* 1. Background-subtraction Algorithm

The first algorithm considered in the new approach was a Background-subtraction method. The method compared consecutive image frames. A Technical Interface (TI) was created to adjust image and movement detection settings as shown in figure 2.

The UI had 8 track-bars used to modify image and movement detection settings:

* Display Video: Display detected movement on a display screen.
* Motion Speed: The number of consecutive frames containing motion.
* Sensitivity: The minimum absolute difference for a given pixel in two consecutive frames to be identified as changed.
* Height in Pixels: Height of image in pixels.
* Width in Pixels: Width of image in pixels.
* Frames per Second: The number of captured images per second.
* Detection Area in Pixels: The minimum number of adjacent changed pixels in two consecutive frames required to be identified as changed.
* Switch Activation Time: This track-bar allowed the new approach to function in two different modes: Switch mode and Time-Delay mode. Setting the Switch Activation Time to 0 would allow the system to function in switch mode, where the system would activate the relay when movement was detected and deactivate it when no movement was detected. Setting the Switch Activation Time to a value other than 0, would allow the system to function in Time-Delay mode, where if a movement was detected, the system would activate the relay for the value set by the track-bar (in seconds), then the relay would be deactivated if no further movement was detected.

Once track-bars were set to the desired values and Apply Settings button was clicked, the approach stored all values in a CSV file. These settings were used to detect movement for different users.

When the Start button from the UI was pressed, the approach used the settings stored in the CSV file and Background-subtraction to compare each pixel of the new frame with the previous frame. If a pixel in the new frame was changed by a value more than the Sensitivity value and a number of adjacent pixels identified as changed were larger than the Detection Area, the system would draw a green contour around the changed pixels and would identify the new frame as changed as shown in figure 3. Where figure 3a shows a user thumb when no movement was detected and figure 3b shows the user thumb when movement was detected and green contours identifying movement area.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Figure 2.** TI used to control Background-subtraction algorithm settings. |  | **Figure 3.** Video feed showing user thumb; a: no movement detected, b: movement detected. |

If the number of consecutive frames identified as changed were higher than the Motion Speed, the program would send logic high voltage to a specific pin. That logic high value would activate the relay.

* 1. Python Imaging Library Algorithm

A second Computer-Vision algorithm was used to improve sensitivity and accuracy. A similar procedure used in the previous Sub-section was followed. Python Imaging Library (PIL) was used instead of Background-subtraction algorithm.

PIL compared each pixel of the new image with the previous image taken. If the new image was different from the previous image, the approach a high logic voltage to the output pin. The high logic value was used to trigger a relay.

Sensitivity to movement and the amount of movement required to trigger the relay could be adjusted using “Sensitivity” and “Threshold” parameters in the Python program. Sensitivity represented the number of pixels in the new image required to be different in order to detect a movement. Threshold represented the level of difference in the same pixels in two consecutive images to be considered as different.

* 1. Open-source-Computer-Vision Algorithm

The third attempt to improve sensitivity and accuracy considered an Open-Source-Computer-Vision (Open-CV) Algorithm. A similar procedure used in the previous Sub-section was followed. OpenCV was used to analyse consecutive images captured. Sensitivity and Detection Area values were assigned. OpenCV algorithm compared each pixel in the new image taken against pixels from previous image. If a pixel in the new image was changed by a value greater than the Sensitivity value and number of neighbouring pixels marked as changed were greater than the Detection Area, the system identified these pixels by drawing a green contour around them.

A new TI was created to adjust system sensitivity and the amount and duration of movement required to trigger the relay circuit. Figure 4 shows the new TI. The new TI had two buttons: Apply Settings and Exit. Four track-bars used to modify OpenCV settings:

* Duration of Motion: The number of consecutive images having motion.
* Sensitivity Threshold: The minimum difference required in a given pixel in two consecutive images to be marked as changed.
* Detection Area in Pixels: The area of neighbouring pixels marked as changed in two consecutive images.
* Switch Activation Time in Seconds: This track-bar permitted the new approach to work in two different functions: Switch and Time-Delay. Setting this track-bar to 0, the new system would operate as a switch (Switch mode), where the relay would be triggered once a movement was detected and deactivated when no movement was detected. Setting this track-bar to any other value, would operate the system in a Time-Delay mode. When a movement was detected, the relay would remain triggered for that value of time (in seconds). Then the relay would be triggered back to off when no movement was detected.

Once track-bars were set to desired values and the Apply Settings button was clicked, the program would store the new values and update values used in program. Figure 5 shows the new system detecting movement with different settings. Figure 5a shows the new system detecting movement with relatively low sensitivity and a large detection area, figure 5b with medium sensitivity and medium detection area and figure 5c with relatively high sensitivity and small detection area. These settings allowed the new system to be used by different users with different levels of functionality and types of disability.

|  |
| --- |
|  |

|  |
| --- |
| **Figure 4.** New TI used to modify OpenCV sensitivity and amount and duration of movement required. |

|  |
| --- |
|  |

|  |
| --- |
| **Figure 5.** New approach detecting movement with different settings: a: low sensitivity and a large detection area, b: medium sensitivity and medium detection area and c: high sensitivity and small detection area. |

1. Discussion and Results

This paper presented a new approach to steer a powered wheelchair using zero-force sensing and Computer-Vision algorithms. The new approach used a circuit that connected a camera, a Raspberry Pi and a relay. A Python program was created and installed on to the Raspberry Pi. The program implemented three Computer-Vision algorithms to conduct image processing and controlled the camera and the relay.

The camera was directed toward a user thumb/limb which generated voluntary movements that were used to control a powered wheelchair. A simple UI was used to operate the new approach and a TI was used to adjust image and movement detection settings. The level of sensitivity, speed and amount of movement required could be adjusted using track-bars in the TI. All three algorithms successfully detected all movements in front of the camera and surrounded it by green contour. The new approach analysed that movement and compared it to stored settings. If change detected in an image was greater than stored values, the new approach would identify that change and trigger a relay used to control a powered wheelchair.

The new approach was tested practically and it successfully detected movement. The new approach operated in silence and did not generate clicking sound when operated.

The new approach could be used by multiple users. Specific values for “Sensitivity” and “Threshold” could be allocated for each new user according to their level of functionality using the trackbars in the TI.

1. Conclusions and Future Work

The new approach used a friendly User Interface, detected movements used to operate a powered wheelchair and needed less effort to operate a powered wheelchair.

To reduce cost, the authors are planning to upload the program and schematic diagram to an open-access platform. Users will be able to download them free of charge.

Three different Computer-Vision algorithms were compared to improve sensitivity to movement and detection accuracy. Results showed that the new approach provided fast movement detection. Track-bars used in the new approach provided enhanced movement detection settings.

Speed of movement considered in this approach could be used to filter out unwanted movement including user hand tremor or undesired movement generated from driving a powered wheelchair on unsettled ground.

OpenCV provided more accurate and sensitive image detection when compared to other Computer-Vision algorithms. PIL was easier to install and setup on to the Raspberry Pi than other algorithms. The Background-subtraction algorithm approach provided faster movement detection.

Future work will further investigate general shifts in impairment from purely physical to more

complex mixes of cognitive/physical. That will be addressed by considering levels of functionality rather

than disability. New transducers and controllers that use dynamic inputs rather than static or fixed inputs will be investigated. Different AI techniques will be investigated and combined with the new types of controllers and transducers to interpret what users want to do. Smart Inputs that detect sounds and dynamic movement of body-parts using new contactless transducers and/or brain activity using EEG

caps will be considered. The use of mixes of EEG, pre-processing, Fourier transform, and wavelet transforms will be investigated to determine user intentions.

Acknowledgments

This research was funded by an EPSRC EP/S005927/1 project titled “Using artificial intelligence to share control of a powered-wheelchair between a wheelchair user and an intelligent sensor system”. Investigators: Sanders, D and Gegov, A. Senior Researchers Haddad M and Langner M.

1. References

[1] Sanders D and Gegov A 2018 Using artificial intelligence to share control of a powered-wheelchair between a wheelchair user and an intelligent sensor system *EPSRC Project, 2019*

[2] Haddad M and Sanders D 2020 Deep Learning architecture to assist with steering a powered wheelchair *IEEE Trans. Neur. Sys. Reh*. **28** 12 pp 2987-2994

[3] Krops L, Hols D, Folkertsma N, Dijkstra P Geertzen J and Dekker R 2018 Requirements on a community-based intervention for stimulating physical activity in physically disabled people: a focus group study amongst experts *Disabil. Rehabil.* **40** 20 pp 2400-2407

[4] Bos I, Wynia K, Almansa J, Drost G, Kremer B and Kuks J 2019 The prevalence and severity of disease-related disabilities and their impact on quality of life in neuromuscular diseases *Disabil. Rehabil.* **41** 14 pp 1676-1681

[5] Frank A and De Souza L 2018 Clinical features of children and adults with a muscular dystrophy using powered indoor/outdoor wheelchairs: disease features, comorbidities and complicationsof disability *Disabil. Rehabil.* **40** 9 pp 1007-1013

[6] Sanders D, Langner M and Tewkesbury G 2010 Improving wheelchair‐driving using a sensor system to control wheelchair‐veer and variable‐switches as an alternative to digital switches or joysticks *Ind Rob: An int' jnl.* **32** 2 pp157-167

[7] Sanders D, Haddad M and Tewkesbury G 2021 Intelligent control of a semi-autonomous Assistive Vehicle *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 613-621

[8] Haddad M, Sanders D, Tewkesbury G and Langner M 2021 Intelligent User Interface to Control a Powered Wheelchair using Infrared Sensors,” *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 640-649

[9] Sanders D and Bausch N 2015 Improving steering of a powered wheelchair using an expert system to interpret hand tremor *Proc. of the International Conference on Intelligent Robotics and Applications, (Portsmouth*) pp 460-471

[10] Sanders D 2016 Using self-reliance factors to decide how to share control between human powered wheelchair drivers and ultrasonic sensors *IEEE Trans. Neur. Sys. Rehab.***25** 8 pp 1221-1229

[11] Haddad M, Sanders D, Gegov A, Hassan M, Huang Y and Al-Mosawi M 2019 Combining multiple criteria decision making with vector manipulation to decide on the direction for a powered wheelchair *Proc. of SAI Intelligent Systems Conference, (London)* pp 680-693

[12] Haddad M, Sanders D, Langner M, Ikwan F, Tewkesbury G and Gegov A 2020 Steering direction for a powered-wheelchair using the Analytical Hierarchy Process *Proc. 2020 IEEE 10th International Conference on Intelligent Systems-IS* (*Varna)* pp 229-234

[13] Haddad M, Sanders D, Thabet M, Gegov A Ikwan F, Omoarebun P, Tewkesbury G and Hassan M 2020 Use of the Analytical Hierarchy Process to Determine the Steering Direction for a Powered Wheelchair *Proc. of SAI Intelligent Systems Conference (London)* pp 617-630

[14] Haddad M and Sanders D 2019 Selecting a best compromise direction for a powered wheelchair using PROMETHEE *IEEE Trans. Neur. Sys. Rehab.* **27** 2 pp 228-235

[15] Haddad M, Sanders D, Ikwan F, Thabet M, Langner M and Gegov A 2020 Intelligent HMI and control for steering a powered wheelchair using a Raspberry Pi microcomputer *Proc. 2020 IEEE 10th International Conference on Intelligent Systems-IS (Varna)* pp 223-228

[16] Haddad M, Sanders D, Langner M, Bausch N, Thabet M, Gegov A, Tewkesbury G and Ikwan F 2020 Intelligent control of the steering for a powered wheelchair using a microcomputer *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 594-603.

[17] Haddad M, Sanders D, Langner M, Omoarebun P, Thabet M and Gegov A 2020 Initial results from using an intelligent system to analyse powered wheelchair users’ data *Proc. of the 2020 IEEE 10th International Conference on Intelligent Systems-IS (Varna*) pp 241-245

[18] Haddad M, Sanders D, Langner M, Thabet M, Omoarebun P, Gegov A, Bausch N and Giasin K 2020 Intelligent system to analyze data about powered wheelchair drivers *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 584-593

[19] Sanders D, Haddad M, Tewkesbury G, Bausch N, Rogers I and Huang Y 2020 Analysis of reaction times and time-delays introduced into an intelligent HCI for a smart wheelchair *Proc. of the 2020 IEEE 10th International Conference on Intelligent Systems-IS (Varna)* pp 217-222

[20] Sanders D, Haddad M, Langner M, Omoarebun P, Chiverton J, Hassan M, Zhou S and Vatchova B 2020 Introducing time-delays to analyze driver reaction times when using a powered wheelchair *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 559-570

[21] Sanders D, Haddad M, Tewkesbury G, Gegov A and Adda M 2021 Are human drivers a liability or an asset?” Proc. of SAI Intelligent Systems Conference (Amsterdam) pp 805-816

[22] Sanders D, Gegov A, Haddad M, Ikwan F, Wiltshire D and Tan YC 2018 A rule-based expert system to decide on direction and speed of a powered wheelchair *Proc. of SAI Intelligent Systems Conference (London)* pp 822-838

[23] Sanders D, Haddad M, Tewkesbury G, Thabet G, Omoarebun P and Barker T 2020 Simple expert system for intelligent control and HCI for a wheelchair fitted with ultrasonic sensors *Proc. of the 2020 IEEE 10th International Conference on Intelligent Systems-IS (Varna*) pp 211-216

[24] Tewkesbury G, Lifton S, Haddad M and Sanders D 2021 Facial recognition software for identification of powered wheelchair users *Proc. of SAI Intelligent Systems Conference (Amsterdam*) pp 630-639

[25] Haddad M, Sanders D, Langner M and Tewkesbury G 2021 One Shot Learning Approach to Identify Drivers *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 622-629

[26] Lewis C 2007 Simplicity in cognitive assistive technology: a framework and agenda for research *Universal Access in the Information Society* **5** 4 pp 351-361