IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 5, May, 2026, pp. 1535-1547
Journal homepage: http://iieta.org/journals/isi

Optical-Flow–Based Head-Gesture Control for a Low-Cost Embedded Smart Wheelchair

Ni'matul Ma'muriyah[1*], Andik Yulianto[1], Agustinus Eko Setiawan[2], Haeruddin[1], Jan Putra Bahtra ASP[3]

[1] Faculty of Computer Science, Universitas Internasional Batam, Batam 29426, Indonesia
[2] Department of Informatics Engineering, Faculty of Technology and Informatics, Aisyah University, Lampung 35372, Indonesia
[3] Electrical Engineering, Faculty of Industrial Technology, Universitas Internasional Batam, Batam 29426, Indonesia

Corresponding Author Email: nimatul@uib.ac.id

https://doi.org/10.18280/isi.310512

Received: 26 January 2026
Revised: 1 April 2026
Accepted: 20 April 2026
Available online: 31 May 2026

Keywords:
optical flow, head gesture recognition, smart wheelchair, embedded assistive technology, Lucas–Kanade method, human–machine interface, computer vision

ABSTRACT

Smart wheelchair interfaces must be accurate, non-invasive, and computationally feasible on low-cost hardware. This study presents an optical-flow–based approach for head-gesture control of a smart wheelchair using a Raspberry Pi 2 for vision processing and an Arduino Uno for motor control. Head motion was segmented in the YCbCr colour space and tracked using Lucas–Kanade optical flow. Horizontal and vertical motion vectors were temporally accumulated and compared with empirically derived thresholds to distinguish intentional gestures from small fluctuations. Five participants performed right, left, up, down, and centre-position gestures under low (80 lux), normal indoor (250 lux), and bright (500 lux) lighting conditions. The evaluation comprised 400 static-condition samples and 160 samples collected during wheelchair motion. Under static conditions, the system achieved an overall gesture-recognition accuracy of 93.8% ± 2.7. During dynamic wheelchair operation, command-execution accuracy was 82.0% ± 5.4, with lower performance attributed to vibration and motion-induced segmentation instability. Recognition accuracy increased from 84.6% under low lighting to 96.2% under bright lighting, while mean response delay ranged from 0.784 to 1.370 s. The results show that accumulated optical-flow features can support lightweight head-gesture control on embedded hardware. Further work should validate the system with a larger and more diverse user group, including intended users, and improve robustness to complex backgrounds, rapid head movements, and low illumination.

1. INTRODUCTION

Mobility is a fundamental aspect of human quality of life because it directly supports individual independence, social participation, and productivity [1]. For individuals with severe physical disabilities, particularly those with quadriplegia or neuromuscular disorders, mobility limitations not only affect daily activities but also restrict access to education, employment, and social life [2]. This condition requires assistive devices that not only support movement but are also able to adapt to the user's residual abilities in a safe and intuitive way [3]. This aspect becomes increasingly crucial in the productive-age group, given that mobility limitations directly affect productivity levels and work independence [4].

One of the most commonly used assistive technologies to support mobility is the wheelchair, which is designed to assist individuals with limited motor function [5]. According to the World Health Organization (WHO), approximately 1.3 billion people worldwide (±16% of the global population) experience significant disabilities, with the increase influenced by population aging and the rising prevalence of non-communicable diseases. WHO also estimates that around 80 million people worldwide (±1% of the global population) require wheelchairs for mobility, yet access to appropriate wheelchairs remains highly uneven, particularly in low- and middle-income countries [6]. Although they improve basic mobility, conventional wheelchairs still rely on user effort or external assistance, thereby limiting user independence and participation in daily activities [7]. Therefore, the development of adaptive wheelchairs represents a promising solution to improve users' quality of life through control systems personalized to individual needs and abilities [7, 8].

Research on smart wheelchairs has significantly increased by integrating mobile robotics and HMI to enhance usability for individuals with disabilities [9]. Haddoun et al. [8] showed that head-movement-based wheelchair control with a non-invasive interface can provide more intuitive navigation for users with severe motor impairments. A similar approach was also developed by Dey et al. [10], who used head movement direction as navigation commands for smart wheelchairs and reported improved user independence. Most of these approaches rely on computer vision to recognize head gestures through facial feature detection or head pose estimation as the basis for decision-making [11].

Most reported head-gesture-based smart wheelchair systems still depend on additional wearable sensors or complex system architectures, such as accelerometers, Inertial Measurement Units (IMUs), or multimodal sensor

combinations [12, 13]. Dependence on external sensors increases cost, implementation complexity, and the need for individual calibration, which potentially limits system deployment in resource-constrained environments and embedded systems [14]. In this context, Optical Flow becomes an attractive method because it can detect motion patterns directly from pixel intensity changes between frames, without requiring complex models or additional sensors, and is relatively computationally lightweight [15, 16].

Although Optical Flow has been widely used in motion detection for various computer vision applications, including head movement tracking and body orientation estimation in Human–Machine Interface (HMI) systems, its specific application for detecting head movements as control commands for smart wheelchairs in low-cost embedded systems remains relatively limited in recent scientific literature. Recent vision-based smart wheelchair studies have utilized head gesture recognition through computer vision approaches such as Haar Cascade–based face and nose detection [11]. However, these approaches still depend on facial feature detection stability and fixed rule-based classification, making them sensitive to environmental variations and less adaptive under dynamic conditions.

Several studies have also demonstrated that Optical Flow algorithms are effective for capturing spatio-temporal motion changes and real-time motion estimation in embedded computer vision systems [15, 16]. Nevertheless, these approaches are generally focused on generic motion estimation or object tracking rather than adaptive head gesture recognition for real-time smart wheelchair navigation under explicit low-computational constraints.

In addition, challenges related to detection stability under varying head movement speeds, adaptation to lighting and background variations, and computational efficiency limitations on embedded devices remain insufficiently addressed in current Optical Flow–based navigation control systems. Based on these research gaps, this study proposes an Adaptive Optical Flow–Based Head Gesture Control approach as a simple, non-invasive, and cost-effective vision-based smart wheelchair control mechanism. The proposed system is designed to operate in real time on resource-constrained computing devices without requiring additional wearable sensors, thereby meeting the requirements of low-resource and cost-efficient embedded hardware. This approach is expected not only to enhance the independence of users with motor impairments but also to expand access to adaptive and affordable assistive technologies for populations with limited mobility. The structure of this paper is organized as follows: Section 2 presents a review of related studies. Section 3 describes the proposed method for optical flow estimation. Section 4 presents the experimental results of the approach implemented in an embedded system. Finally, Section 5 presents the conclusions of this study.

2. RELATED WORK

This section presents a review of previous studies relevant to the development of head-gesture-based smart wheelchairs. The discussion focuses on the development of adaptive wheelchair technology, computer vision–based HMI approaches, and the use of Optical Flow methods for head movement detection. This review aims to identify research trends, the strengths and limitations of existing approaches, and research gaps that underlie the proposal of the Adaptive Optical Flow–Based Head Gesture Control method in this study.

Previous studies indicate that head-movement-based smart wheelchair control is an effective solution for enhancing the independence of users with severe motor disabilities. Haddoun et al. [8] and Dey et al. [10] reported that inertia- and accelerometer-based approaches can achieve high accuracy, but still depend on wearable devices and static thresholds, which limit system adaptability to user variations and real-world environmental conditions. In addition, the use of additional sensors and relatively complex system architectures poses challenges for low-cost embedded implementations.

On the other hand, computer vision–based approaches offer a more non-invasive solution. Abdulkareem et al. [12] showed that facial landmark tracking can achieve very high accuracy in detecting head movements, but still relies on long movement durations, fixed thresholds, and exhibits sensitivity to variations in lighting and environmental conditions. Meanwhile, Ammar et al. [15] demonstrated that Lucas–Kanade–based Optical Flow with pyramidal refinement is effective for real-time motion estimation in embedded systems; however, this approach is designed for general motion estimation and still uses static parameters, making it less optimal for gesture-based control applications under resource constraints.

Based on these limitations, this study proposes an Adaptive Optical Flow approach for head-movement-based smart wheelchair control, which is designed to be computationally lightweight, adaptive to user movement dynamics and environmental conditions, and does not require additional wearable devices. This approach is expected to bridge the gap between high accuracy, user comfort, and implementation feasibility in embedded systems with constrained resources.

More specifically, the proposed approach is optimized for head gesture control in smart wheelchairs, with mechanisms that adapt to user movement dynamics and environmental conditions in real time. The main contribution of this study lies in the integration of computationally lightweight vision-based motion estimation with an adaptive scheme, enabling a balance between accuracy, responsiveness, environmental robustness, and implementation feasibility in resource-constrained embedded systems, an aspect that has not been explicitly addressed in previous studies.

Based on the comparison presented in Table 1, recent smart wheelchair studies have mainly focused on wearable sensors, Haar Cascade–based detection, and deep learning–based gesture recognition approaches. Although these methods demonstrate promising performance, several limitations remain, including dependence on wearable devices, high computational complexity, sensitivity to environmental variations, and limited discussion regarding lightweight embedded implementation.

In contrast, the proposed study introduces an Adaptive Optical Flow–based head movement recognition approach implemented on a Raspberry Pi 2 and Arduino Uno platform. The proposed method integrates motion vector accumulation and adaptive thresholding to improve head movement recognition under both static and dynamic conditions without requiring wearable sensors or computationally expensive deep learning models. This approach provides a lightweight, low-cost, and non-invasive smart wheelchair control system suitable for real-time embedded assistive applications.

Table 1. Comparison of recent studies on vision- and gesture-based smart wheelchair control

Study	Control Input	Method	Platform/Implementation	Main Strength	Limitation
Somawirata and Utaminingrum [11]	Head gesture	Haar Cascade–based face and nose detection using computer vision	Vision-based smart wheelchair control	Non-invasive head gesture control without wearable sensors	Accuracy decreases as camera distance increases and depends on stable face/nose detection
Othman and Yuniarno [17]	Hand gesture	CNN with MediaPipe-based spatial feature extraction	Camera-based smart wheelchair with ESP32 and Wi-Fi control	High gesture classification performance and real-time control capability	Requires large training data and computational resources; FPS still relatively low (~6 FPS)
Nguyen et al. [18]	Hand gesture	Vision-based hand gesture recognition using YOLOv8n and MediaPipe	Smart wheelchair system with Jetson Nano embedded platform	High gesture recognition accuracy and effective real-time wheelchair navigation	Focuses on hand gestures and still requires relatively higher computational resources for embedded deployment
Haddoun et al. [8]	Head motion	IMU-based head gesture control	Arduino Nano + Bluetooth electric wheelchair	Suitable for severe motor impairment with high directional accuracy	Requires wearable motion-sensing hardware and head-mounted sensors
Abdulkareem et al. [12]	Head movement	Real-time head movement detection	Intelligent wheelchair control system	Enables responsive real-time wheelchair navigation	Performance affected by subtle head movements and embedded processing limitations
Proposed study	Head gesture	Adaptive Optical Flow with vector accumulation and thresholding	Raspberry Pi 2 + Arduino Uno	Lightweight, low-cost, non-invasive embedded implementation without wearable devices	Sensitive to lighting changes and excessive dynamic motion

Note: CNN = Convolutional Neural Network; IMU = Inertial Measurement Unit; FPS = Frames Per Second.

3. MATERIALS AND METHODS

The research method began with a literature review of relevant previous studies, followed by the selection of appropriate methods to achieve the research objectives. The object of this research was a smart wheelchair system controlled through the detection of user head movements using the Optical Flow method. Figure 1 presented the system block diagram, illustrating the process from head movement input to its translation into smart wheelchair motion.

The Optical Flow method was used to estimate pixel motion vectors between consecutive images acquired in real time by the camera, thereby representing the dynamic patterns of user head movements. Based on the characteristics of the generated vectors, the system identifies the intended direction of head movement.

The head movement information was subsequently mapped into a DC motor control scheme to generate appropriate rotation commands, which enabled g the wheelchair to maneuver responsively.

The design was divided into two parts: hardware design, consisted of a wheelchair, DC motors as actuators, a camera as a sensor, a Raspberry Pi 2, and a motor controller. The software design involved optical image processing using Optical Flow to recognize head movements. Figure 2 presented the hardware system flow. The hardware design consisted of a webcam, a Raspberry Pi 2, an Arduino Uno, a motor controller, and DC motors.

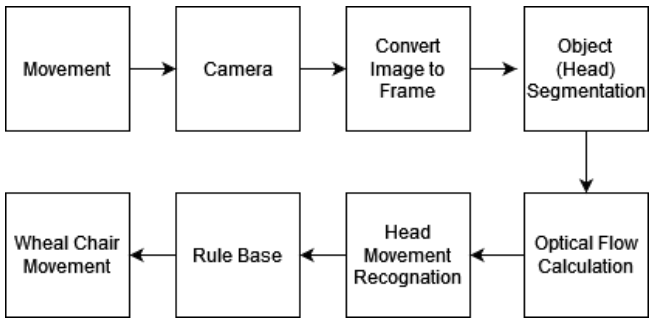

Figure 1. System block diagram

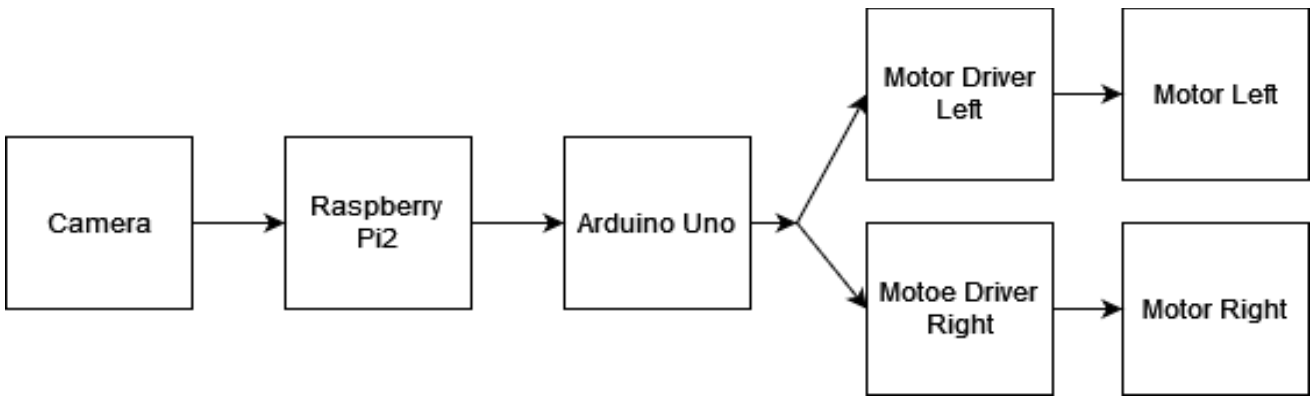

Figure 2. Hardware system flow

Figure 2 showed that the hardware design consisted of three blocks: (a) an image processing block generated by the webcam and Raspberry Pi 2; (b) an image-to-motor motion conversion block that used an Arduino Uno and a motor controller module; and (c) a wheelchair mechanism that consisted of the wheelchair as the main system and DC motors as actuators.

The first discussion focused on the image processing unit. Based on Figure 1, the main system input consisted of digital images of the user's head movement, therefore, the design of the image processing unit was considered a crucial component of the system. The Raspberry Pi 2 was selected as the visual sensor based on the need for mini-computer-level computation capabilities to perform real-time image processing. Literature review results indicate that the Raspberry Pi 2 has adequate performance for image processing applications [19, 20].

This system used a Logitech C525 webcam as a digital image acquisition device. The webcam was equipped with an 8-megapixel sensor, supported video resolutions of up to 720p with H.264 encoding, and utilized a USB interface to transfer image data to the Raspberry Pi 2. To ensure optimal acquisition of the user's head movements, the webcam was positioned directly in front of the user without obstruction, thereby allowing the head movements to be recorded clearly and consistently.

The next block was related to converting the image processing results into motor motion, and was implemented using an Arduino Uno and a motor controller module. The Arduino Uno functioned as the main controller that generated Pulse Width Modulation (PWM) signals to regulate motor speed and rotation direction through the motor controller module. The module used was the Elechouse 20 A Single Driver Module, which was selected because it could handle currents of up to 20 A and supported precise control of motor direction and speed. The control signal rules used to generate motor motion were presented in Table 2.

Table 2. Motor driver control signal configuration

State	A	K1	K2	Out1	Out2
Forward Speeding	H	PWM	H	~ PWM	L
Reverse Speeding	H	H	PWM	L	~ PWM
Brake	H	H	H	L	L
Brake	L	X	X	L	L

Signal isolation between the control signals and motor signals was implemented using an optocoupler integrated into the motor controller module, thereby protecting logic control units such as the Arduino Uno from electromagnetic interference and current surges generated by the motor. This controller module used four MOSFETs configured as a full-bridge driver, exhibited low resistance characteristics, and supported PWM pulse-width control in the range of 0%–98% at operating frequencies between 500 Hz and 100 kHz. Based on the control rules presented in Table 2, the motor could be operated in forward, reverse, or braking modes.

The wheelchair system was designed using two DC motors connected to the left and right wheels, requiring two single-channel motor controller modules. The Arduino Uno functioned as the main controller that generated PWM control signals and independently transmitted them to each motor controller module to regulate rotation direction and speed of the motors.

Figure 3 showed the mechanical structure of the wheelchair, which was a modified version of a conventional wheelchair with GM-42 DC motors installed on each wheel. The selection of the mechanical design and motor type is based on a literature review and considerations of the motor's capability to support and move the wheelchair's operational load stably.

Figure 3. Design of the wheelchair

The software design consisted of two main components: software implemented on the Arduino Uno for motor control and software implemented on the Raspberry Pi 2 for image processing based on the Optical Flow method. The Arduino Uno received motion commands from the Raspberry Pi 2 through serial communication in the form of ASCII codes and subsequently generated control signals to control the direction and speed of the motors. The motor rotation rules employed in the differential drive scheme were presented in Table 3.

Table 3. The motor rotation rules on wheelchair movement

ASCII	Movement	Right Motor Rotation	Left Motor Rotation
A	Forward	Clockwise	Counter clockwise
B	Backward	Counter clockwise	Clockwise
C	Right Rotation	Clockwise	Clockwise
D	Left Rotation	Counter clockwise	Counter clockwise
E	Stop	Stop	Stop

By integrating the control signal rules on the motor driver module presented in Table 2 with the motor rotation rules for smart wheelchair movement presented in Table 3, a configuration defining the relationship between motor drive signals and smart wheelchair movement patterns was obtained. This configuration was systematically formulated and was presented in detail in Table 4.

Table 4. The motor driver configuration with intelligent wheelchair movements

ASCII	AR	K1R	K2R	AL	K1L	K2L
A	H	H	PWM	H	PWM	H
B	H	PWM	H	H	H	PWM
C	H	H	PWM	H	H	PWM
D	H	PWM	H	H	PWM	H
E	L	X	X	L	X	X

As shown in Table 4, the ASCII symbols A, B, C, D, and E represented the movement commands defined in Table 2, where A denoted forward motion, B denoted backward motion, C denoted right rotation, D denoted left rotation, and E denoted stop. This representation was employed to simplify the transmission of motion commands from the Raspberry Pi 2 to the Arduino Uno via serial communication.

The image processing stages were divided into two main processes, namely (a) the segmentation process and (b) the head movement recognition process. The output of these two processes were used as the basis for determining wheelchair control commands. The flowchart of the head object segmentation process was shown in Figure 4. The segmentation process served as an initial stage for identifying the presence and region of the user's head. Subsequently, the segmentation results were analyzed using the Optical Flow method to determine the direction of movement and the threshold values of head movement.

In this study, the segmentation process was performed using the YCbCr color space to separate the head object from the background. The skin color value ranges used were within the intervals of 77 < Cr < 127 and 133 < Cb < 173. The complete algorithm of the head object segmentation process was presented in Figure 4.

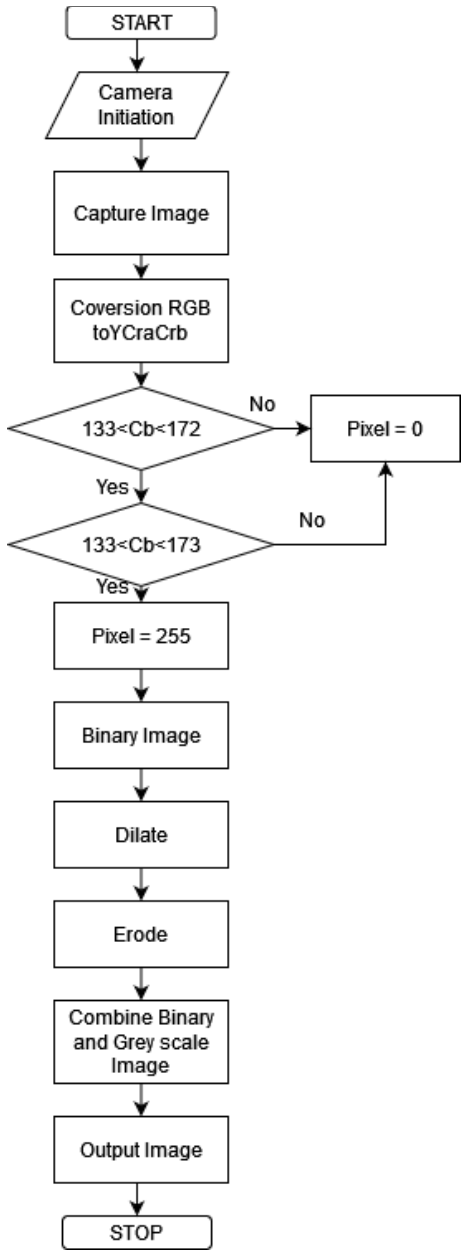

Figure 4. Flowchart of the head object segmentation process

Figure 4 showed the flowchart of the head object segmentation process. The segmentation results were represented in grayscale images. As shown in Figure 4, the output of this stage was a motion vector representing changes in pixel values associated with object movement. The motion vector consisted of two primary components: displacement along the horizontal (x) axis and displacement along the vertical (y) axis. This vector information is then used to determine the direction and magnitude of head object displacement in the image, which forms the basis for the movement recognition process and smart wheelchair control.

The optical flow between two images can be estimated using the optical flow constraint equation as follows:

$$I_x u + I_y v + I_t = 0 \quad (1)$$

where,
I_x = image intensity gradient in the x-direction
I_y = image intensity gradient in the y-direction
I_t = temporal image intensity derivative
u = horizontal Optical Flow component
v = vertical Optical Flow component
where, u and v denote the horizontal and vertical Optical Flow vector components measured in pixels/frame, respectively.

One of the commonly used methods for Optical Flow estimation is the Lucas–Kanade approach. This method performs local motion estimation using pixel intensity variations within a small neighborhood window, enabling efficient and stable detection of object movement between consecutive image frames. Compared with global Optical Flow methods such as Horn–Schunck and multi-scale approaches such as Pyramidal Lucas–Kanade, the standard Lucas–Kanade method generally provides lower computational complexity and faster processing time, making it more suitable for real-time embedded systems with limited hardware resources such as Raspberry Pi 2.

Based on experimental evaluation, the 3 × 3 window configuration was selected because it provided sufficiently stable motion estimation while maintaining lower processing delay compared with larger window sizes such as 5 × 5 and 7 × 7. Therefore, the Lucas–Kanade 3 × 3 configuration was considered the most appropriate for lightweight real-time smart wheelchair navigation.

4. RESULTS AND DISCUSSION

The prototype of the smart wheelchair that uses head movements as a control mechanism for wheelchair motion is shown in Figure 5.

Figure 5. Prototype of smart wheelchair

The operational systems of the smart wheelchair prototype are illustrated in the flowchart in Figure 6.

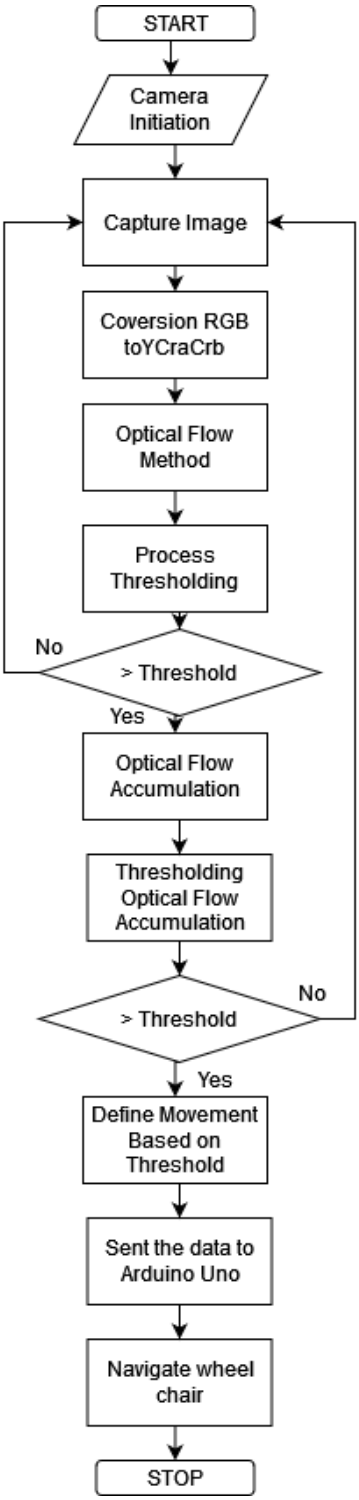

Figure 6. Automatic wheelchair navigation control program flowchart with head movements

4.1 Experimental setup

The experiments were conducted in an indoor environment using the proposed smart wheelchair prototype equipped with a Logitech C525 webcam, Raspberry Pi 2, Arduino Uno, and DC motor controller modules. The experimental evaluation was designed to assess the reliability and stability of the proposed Adaptive Optical Flow–based head gesture recognition system under both static and dynamic operating conditions.

A total of five participants were involved in the experiments, consisting of users aged between 21 and 32 years. Each participant performed five head movement commands, namely turn right, turn left, look up, head down, and return to the center position. Each movement scenario was repeated multiple times to evaluate system consistency and reduce random experimental bias.

The experiments were conducted under three different lighting conditions, namely low lighting (80 lux), normal indoor lighting (250 lux), and bright lighting (500 lux). During testing, the distance between the participant and the webcam was maintained within the range of 60–80 cm to ensure stable head object acquisition. The overall experimental configuration used in this study is summarized in Table 5.

Table 5. Experimental setup and testing configuration

Parameter	Description
Participants	5 participants
Age Range	21–32 years
Head Movement Commands	Right, Left, Up, Down, Center
Repetitions	10 repetitions per movement
Camera Distance	60–80 cm
Lighting Conditions	Low (~80 lux), Normal (~250 lux), Bright (~500 lux)
Experimental Environment	Indoor
Static-condition Samples	400 samples
Dynamic Wheelchair Movement Samples	160 samples
Total Experimental Samples	560 samples

The process began with image acquisition by the webcam, followed by head object segmentation and Optical Flow computation to obtain movement vectors. The optical flow results were then compared with threshold values and accumulated to determine the direction of wheelchair movement. The movement commands were subsequently sent by the Arduino to the motor controller module according to the predefined rules. The rule base of the control system was shown in Figure 7.

In Figure 7, the horizontal Optical Flow component was represented by u, while the vertical component was represented by v. Positive and negative values indicate the direction of head movement in the Cartesian coordinate system. To improve readability and interpretation, the movement directions and control command mappings were clearly labeled in the figure. The generated movement commands were subsequently translated into wheelchair navigation instructions, including forward, backward, left rotation, right rotation, and stop conditions.

Table 6 presented the head movement rules derived from testing the accumulation of optical flow with respect to head orientation angles. The optical flow values were obtained based on Eq. (1), where the values represented the average of the u and v vector components.

The test results shown distinct accumulated Optical Flow patterns for the four primary head movement directions, namely right (u^+), left (u^-), up (v^+), and down (v^-), as illustrated in Figure 8. Positive horizontal movements generate positive accumulated values in the u component, while negative horizontal movements produced negative u values. Similarly, upward head movements generated positive accumulated values in the v component, whereas downward movements produce negative v values.

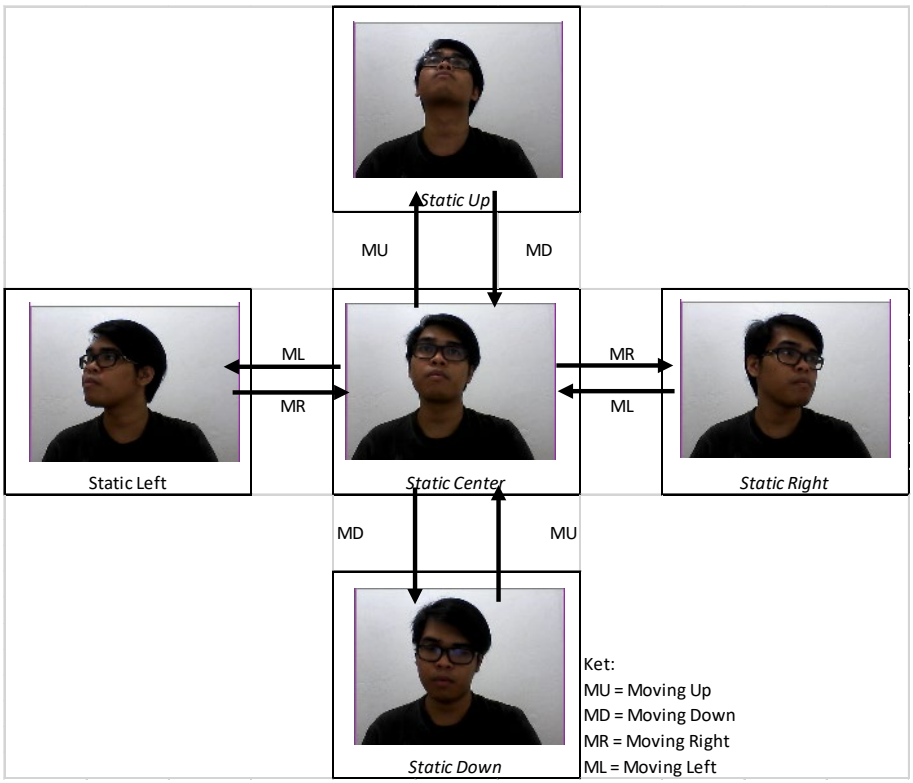

Figure 7. Head movement control rules and command generation process for smart wheelchair navigation

Table 6. Threshold-based rules for head movements

Head Movement Condition	Accumulated Horizontal Optical Flow (u)	Accumulated Vertical Optical Flow (v)
Moving Right	$u_{total} > 3$	0
Moving Left	$u_{total} < -3$	0
Moving Up	0	$v_{total} > 2$
Moving Down	0	$v_{total} < -2$

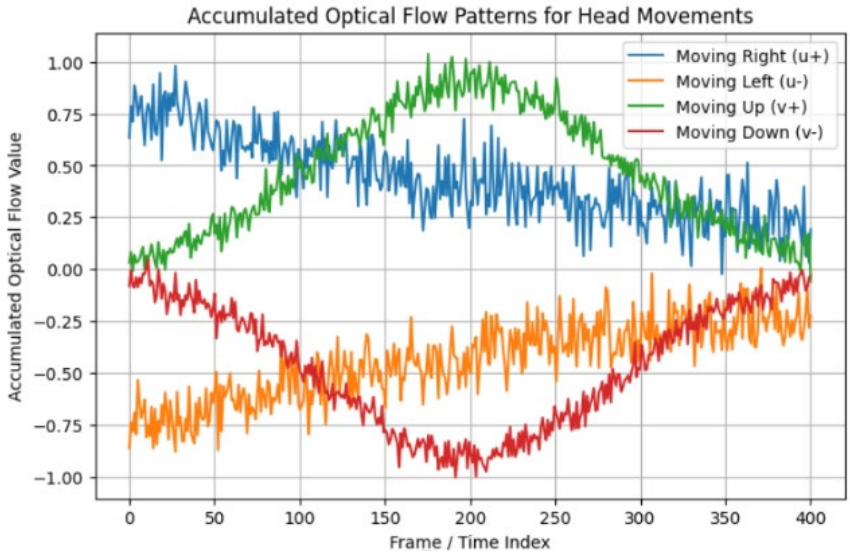

Figure 8. Temporal characteristics of Optical Flow signals for head gesture recognition

The temporal distribution of these accumulated Optical Flow signals demonstrates clear separation between horizontal and vertical movement patterns despite the presence of noise fluctuations. These findings indicate that accumulated Optical Flow vectors provide stable and discriminative motion features for reliable head movement classification in the proposed smart wheelchair control system.

Furthermore, the threshold-based accumulated Optical Flow vector approach was designed to improve system stability against noise interference and unintended small motion fluctuations. Consequently, the system responds only to consistent and significant head movements, thereby enhancing recognition reliability and user comfort during wheelchair operation. Table 7 further summarized the

relationship between head movement directions and the average Optical Flow vector components. Horizontal head movements correspond to displacement along the X-axis, primarily affecting the horizontal Optical Flow component (u), while the vertical component (v) remains relatively stable. Conversely, vertical head movements affected the vertical Optical Flow component (v), whereas the horizontal component (u) tended to remain relatively unchanged.

Table 7. Relationship between head movement direction and average Optical Flow vector components

Head Movement Direction	Horizontal Optical Flow Component, u (pixels/frame)	Vertical Optical Flow Component, v (pixels/frame)	Movement Interpretation
Positive horizontal movement	Positive accumulated values	Approximately constant	Head moves to the right
Negative horizontal movement	Negative accumulated values	Approximately constant	Head moves to the left
Positive vertical movement	Approximately constant	Positive accumulated values	Head moves upward
Negative vertical movement	Approximately constant	Negative accumulated values	Head moves downward

In Table 7, u represented the horizontal Optical Flow vector component, while v represented the vertical Optical Flow vector component. The Optical Flow values were expressed in pixels/frame, representing the average pixel displacement velocity between consecutive image frames. As illustrated in the overall system flowchart in Figure 6, the primary input of the proposed system was image data.

Therefore, the selection of the window size in the Optical Flow estimation processed becomes an important factor in achieving optimal performance. The window size determined the number of neighboring pixels used as references during computation, where larger window sizes generally produced more stable estimations but needed higher computational cost.

Based on experiments using the Lucas–Kanade algorithm, three window size variations were evaluated, namely 3 × 3, 5 × 5, and 7 × 7. The experimental results showed that all window sizes produced relatively similar Optical Flow estimation patterns; however, larger window sizes resulted in longer processing time.

Therefore, a 3 × 3 window size was selected in this study to maintain computational efficiency and support real-time operation on the embedded platform.

Figure 9 presented examples of the segmentation process results, including the original input images, binary segmentation masks, and masked grayscale images used as the initial stage before Optical Flow estimation. The figure also illustrated the influence of environmental lighting conditions on the quality of head object extraction and segmentation consistency prior to motion analysis.

The images, it could be concluded that low lighting conditions and uneven illumination reduce the quality of head object segmentation results. This segmentation imperfection directly affects the Optical Flow estimation process, causing the performance of the head movement recognition system to become suboptimal under low-light conditions. Therefore, the reliability of the system was strongly influenced by the quality and stability of environmental lighting conditions.

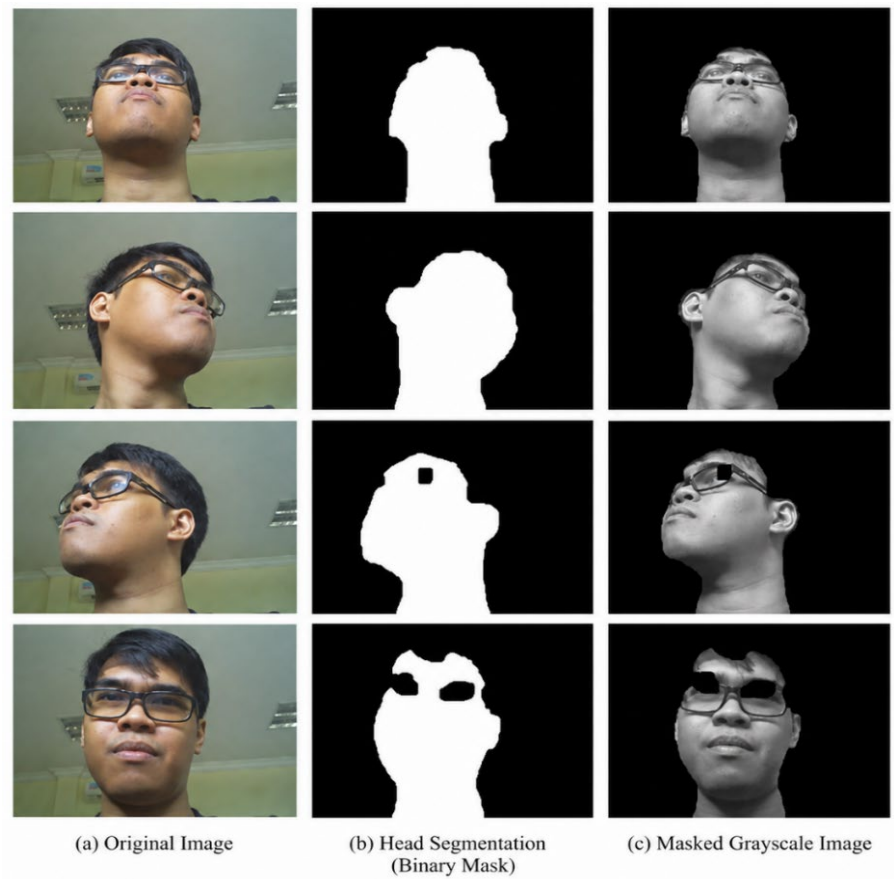

Figure 9. Examples of original images, binary segmentation masks, and masked grayscale images for head movement detection

4.2 Performance evaluation under different lighting conditions

To further evaluate the robustness of the proposed Adaptive Optical Flow approach, additional experiments were conducted under different lighting conditions, namely low lighting (80 lux), normal indoor lighting (250 lux), and bright lighting (500 lux). The experiments aimed to analyze the influence of lighting intensity on head gesture recognition accuracy and system response delay. The evaluation results are summarized in Table 8.

The experimental results demonstrated that lighting conditions significantly influence both recognition accuracy and system response delay. Under bright lighting conditions (500 lux), the proposed system achieved the highest recognition accuracy of 96.2% with the lowest response delay of 0.784 seconds. Under normal indoor lighting (250 lux), the system maintained stable performance with an accuracy of 93.8% and an average delay of 0.982 seconds.

Table 8. Performance evaluation under different lighting conditions

Lighting Condition	Illumination Level (lux)	Recognition Accuracy (%)	Standard Deviation (SD)	Average Response Delay (s)
Bright Lighting	500 lux	96.2	±1.8	0.784
Normal Indoor Lighting	250 lux	93.8	±2.7	0.982
Low Lighting	80 lux	84.6	±4.9	1.370

In contrast, low-light conditions (80 lux) reduced recognition accuracy to 84.6% and increased the response delay to 1.37 seconds. This performance degradation mainly occurred because insufficient illumination reduced segmentation quality and introduced instability in the Optical Flow estimation process. Poor lighting conditions affected the consistency of head object extraction, resulting in unstable motion vector accumulation and increased processing overhead. These findings confirm that environmental lighting was an important factor influencing the reliability, responsiveness, and stability of real-time vision-based embedded control systems.

In this study, head movement determination was based on the concept of particle motion, where object displacement was estimated by aggregating Optical Flow values obtained from several consecutive image frames. These Optical Flow values represent pixel displacement velocity (pixels/frame), while the aggregation process estimates the overall displacement of the head object over time.

The resulting displacement information is subsequently used to determine movement direction and generate wheelchair control commands according to predefined threshold rules. This aggregation-based approach enabled the system to distinguish intentional head movements from small unintended motion fluctuations and image noise. Consequently, the proposed Adaptive Optical Flow method improved the stability and robustness of head gesture recognition, particularly under dynamic operating conditions and varying environmental lighting conditions.

Figure 10 presented the distribution of Optical Flow values under static conditions, namely when the head remained stationary or after transitions between movement directions. The transition from static to moving conditions was detected by applied thresholding to the horizontal and vertical Optical Flow vector components, namely u and v.

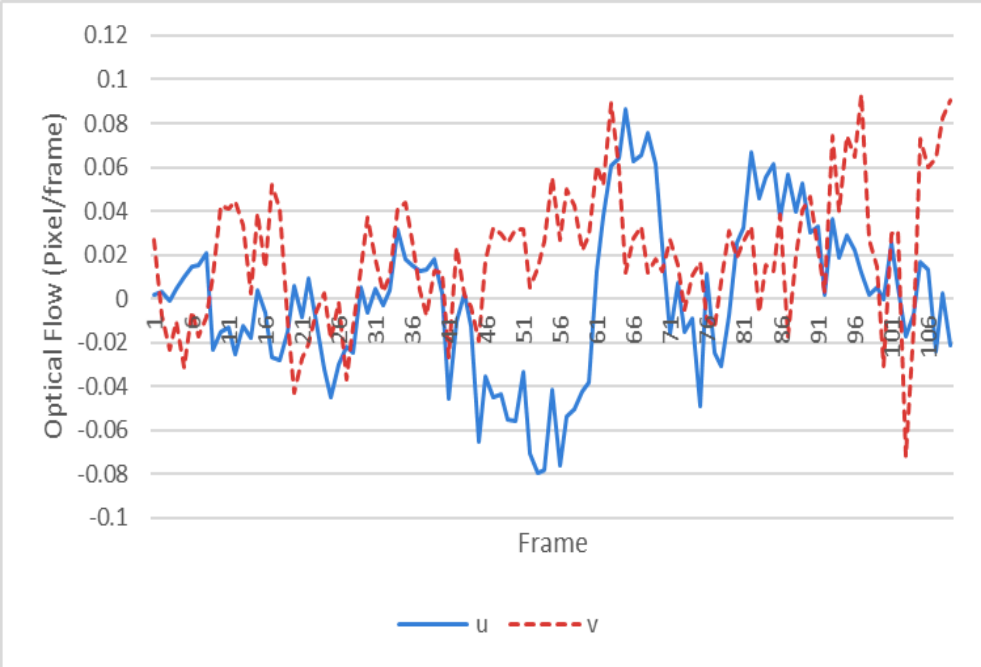

Figure 10. Distribution of Optical Flow vector values under static head conditions

The threshold values were determined empirically through statistical analysis of the average u and v distributions obtained during repeated calibration experiments under static head conditions. Experimental observations showed that under

static conditions, the Optical Flow values generally fluctuated within the interval of approximately -0.1 to 0.1 due to image noise, illumination variations, and small unintended head movements. Therefore, values outside this interval were used as indicators of intentional head movement.

Furthermore, the proposed system applied a temporal accumulation mechanism in which Optical Flow vectors from several consecutive frames were combined before generating movement commands. This mechanism functions as an adaptive temporal filtering approach that improved robustness against transient noise fluctuations and enhances system stability during dynamic wheelchair operation.

Optical Flow accumulation was subsequently performed when the head was detected to be in a moving condition, which occurs when the u and v values exceed the predefined thresholds.

The threshold criteria used to determine the onset of the moving condition were formulated in detail and presented in Table 9.

Table 9. Threshold criteria for static and dynamic head movement detection based on Optical Flow components

Head Condition	Threshold Criteria	Interpretation
Static condition	$-0.1 < u < 0.1$ and $-0.1 < v < 0.1$	No significant head movement detected
Vertical movement	$v > 0.1$ or $v < -0.1$	Upward or downward head movement detected
Horizontal movement	$u > 0.1$ or $u < -0.1$	Leftward or rightward head movement detected

The threshold values were determined empirically through repeated calibration experiments under static head conditions. Values outside the interval indicate significant pixel displacement caused by intentional head movement.

By applying the predefined rules, the Optical Flow accumulation process began when the system detected a moving condition.

After accumulation was performed, head movements were then evaluated through a series of tests. The tests also included variations in movement orientation angles, where the orientation angle affected the magnitude of pixel displacement in the head object. The results of these tests were presented in the following section.

To enable the system to recognize upward, downward, rightward, and leftward movements, thresholding was applied to the accumulated Optical Flow values. The determination of threshold values was based on test data shown in Figure 11 and summarized in Table 6. The threshold value was set higher than the accumulated Optical Flow value at a 15° orientation angle, with the aim of ignoring small unintended movements and reducing potential errors in the head movement recognition process.

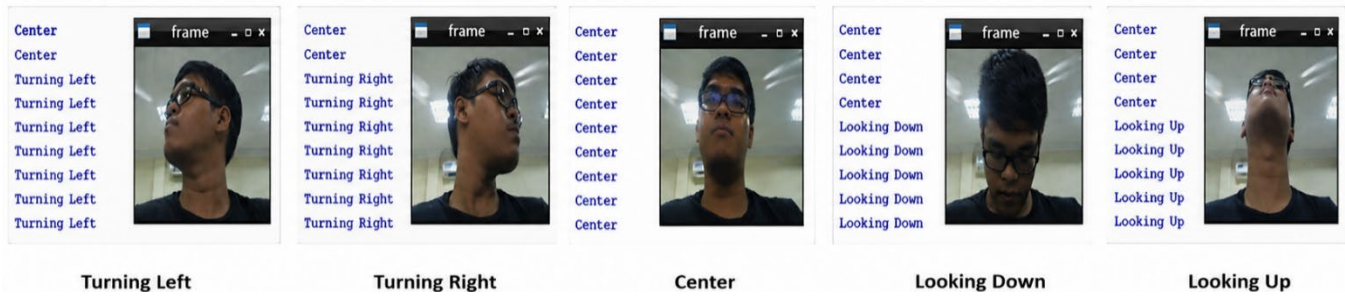

Figure 11. Results of the head movement recognition

Subsequent experiments were conducted to evaluate the success rate and stability of the proposed system in translating user head movements into smart wheelchair control commands under static conditions.

The experiments involved repeated trials under different environmental conditions, including variations in lighting intensity and user movement patterns. The results of the head movement recognition experiments were presented in Table 10. Based on the experimental results, the proposed system demonstrated effective and stable performance in recognizing and implementing head movement commands in accordance with the proposed design. A total of 400 experimental samples were evaluated, resulting in an overall recognition accuracy of 93.8% ± 2.7. The relatively low standard deviation indicated that the system provided consistent recognition performance across repeated trials and different testing conditions.

Despite achieving high recognition accuracy, several recognition errors were still observed. These errors mainly occurred when the accumulated Optical Flow values did not exceed the predefined threshold values. Such conditions were influenced by several factors, including uneven lighting conditions, variations in segmentation quality, and subtle head movements that generated insufficient pixel displacement to satisfy the movement threshold criteria.

Table 10. The results of testing the head movement recognition system with the Optical Flow during static conditions

Head Movement	Number of Tests	Forward (A)	Backward (B)	Right (C)	Left (D)	Stop (E)	Recognition Accuracy (%)
Turn Right	50	48	0	0	0	2	96.0
Turn Left	50	0	46	0	0	4	92.0
Head Down	50	0	0	46	0	4	92.0
Look Up	50	0	0	0	49	1	98.0
Return to Center Position	200	5	4	5	4	182	91.0
Overall Performance	**400**	—	—	—	—	—	**93.8 ± 2.7**

Further experiments were conducted to evaluate the effectiveness of the proposed system in translating recognized head gestures into actual smart wheelchair movements under dynamic operating conditions. Unlike the static-condition experiments, these tests involved real wheelchair motion, which introduced additional challenges such as vibration, environmental disturbances, and motion instability. Table 11 presented the overall system testing results, indicating the alignment between user head movements and wheelchair motion execution.

The experimental results indicated that the wheelchair was able to execute movement commands according to the intended head gestures, achieving an overall execution accuracy of 82.0% ± 5.4. This performance was lower than the recognition accuracy obtained under static conditions, indicated that dynamic wheelchair motion significantly affected system stability. The mismatch between recognized head movements and wheelchair responsed occurred more frequently during wheelchair motion due to vibration, segmentation instability, and Optical Flow accumulation fluctuations.

Table 11. The results of testing the movement of a wheelchair based on head movements

Head Movement Command	Number of Tests	Forward (A)	Backward (B)	Right (C)	Left (D)	Stop (E)	Execution Accuracy (%)
Turn Right	20	1	0	18	0	1	90.0
Turn Left	20	0	1	0	16	3	80.0
Head Down	20	19	0	0	0	1	95.0
Look Up	20	0	17	0	0	3	85.0
Return to Center Position	80	7	8	7	10	48	60.0
Overall Performance	**160**	—	—	—	—	—	**82.0 ± 5.4**

A response delay ranging from 0.784 seconds to 1.37 seconds was also observed during dynamic operation. This delay was influenced not only by Optical Flow image processing, but also by sequential frame acquisition, Optical Flow accumulation across multiple frames, serial communication latency between Raspberry Pi 2 and Arduino Uno, motor response time, and the computational limitations of the embedded platform. Several optimization strategies may improve system responsiveness, including reducing image resolution, optimizing the Optical Flow computation window, minimizing redundant frame processing, and implementing lightweight motion prediction approaches.

In addition, upgrading to more powerful embedded platforms such as Raspberry Pi 4 or NVIDIA Jetson Nano may significantly improve real-time processing performance and reduce system latency. The observed response delay may become a practical limitation in highly dynamic real-world environments where rapid wheelchair response is required for safe navigation. Therefore, reducing system latency remains an important aspect for future development. In addition to algorithm optimization and hardware upgrades, parallel computing approaches such as multithreaded frame acquisition, concurrent image processing, and GPU-accelerated computation on embedded platforms may further reduce processing delay and improve real-time system responsiveness.

Therefore, future work should focus on both algorithm optimization and embedded hardware enhancement to achieve faster and more stable real-time smart wheelchair navigation performance.

The proposed Adaptive Optical Flow framework was designed using computationally lightweight image processing operations, making it portable to more advanced embedded computing platforms. The use of more powerful hardware platforms such as Raspberry Pi 4 or NVIDIA Jetson Nano is expected to improve frame processing speed, reduce response delay, increase system responsiveness, and enhance robustness under dynamic operating conditions. In particular, platforms equipped with higher CPU performance, larger memory capacity, and GPU acceleration may enable higher-resolution image processing, faster Optical Flow computation, and more stable real-time wheelchair navigation performance. Therefore, the proposed system had the potential to be further optimized and scaled for more complex real-world assistive applications using next-generation embedded hardware platforms.

The scalability of the proposed Adaptive Optical Flow system was influenced by several factors, including wheelchair platform characteristics, camera resolution, and sensor update rate. Higher camera resolutions may improve segmentation detail and motion estimation accuracy; however, they also increase computational overhead and processing delay on resource-constrained embedded platforms. Similarly, higher sensor frame rates may improve motion responsiveness and smoother Optical Flow tracking, but require faster real-time processing capability.

In addition, different wheelchair platforms may exhibit different vibration characteristics, motor response times, and mechanical stability, which can affect Optical Flow consistency during navigation. Nevertheless, because the proposed method was based on lightweight image processing operations, the system was expected to remain portable and scalable across different embedded platforms with appropriate parameter adjustment and hardware optimization. Therefore, future work should include scalability evaluation using different wheelchair configurations, camera specifications, and sensor update rates under more diverse operating environments.

The robustness of the proposed Adaptive Optical Flow system was further evaluated under dynamic operating conditions, including rapid head movements and complex backgrounds. Experimental observations showed that rapid head movements reduced recognition accuracy to approximately 84–86%, while complex background conditions reduced performance to approximately 80–82% due to segmentation instability and Optical Flow fluctuations.

Most misrecognition cases occurred when head movements were too fast, too subtle, or when environmental disturbances and uneven lighting reduced segmentation quality. These conditions increased misrecognition rates to approximately 14–18% under dynamic operating conditions.

Although the proposed system still demonstrated stable overall performance for embedded smart wheelchair navigation, these findings indicated that environmental

robustness remained an important challenge for real-time vision-based assistive systems. Therefore, future improvements should focus on adaptive background filtering, segmentation stability, and lightweight motion stabilization mechanisms to improve robustness under more complex real-world environments.

Although formal usability evaluation and subjective user experience analysis were not conducted in this study, experimental observations indicated that participants were generally able to understand and operate the proposed head gesture control mechanism after a short adaptation period. The non-invasive vision-based interaction approach also reduced the need for additional wearable devices, which may improve user comfort during operation. Nevertheless, comprehensive usability evaluation involving user satisfaction, mental workload, comfort, and long-term interaction experience remains necessary for future development and practical deployment of the proposed smart wheelchair system.

5. CONCLUSIONS

This study proposed an Adaptive Optical Flow–Based Head Gesture Control system for smart wheelchair navigation on embedded platforms. Experimental results demonstrated that the proposed system achieved a head movement recognition accuracy of 93.8% under static conditions and an overall wheelchair movement execution accuracy of 82.0% under dynamic operating conditions. The results indicated that the proposed accumulation-based Optical Flow approach was capable of distinguishing horizontal and vertical head movements consistently while maintaining computational feasibility on low-cost embedded hardware such as Raspberry Pi 2 and Arduino Uno.

The experimental evaluation under different lighting conditions also showed that environmental illumination significantly affected segmentation quality, Optical Flow stability, recognition accuracy, and system response delay. Under low-light conditions, recognition performance decreased and response delay increased due to segmentation instability and fluctuations in Optical Flow estimation. In addition, experiments under dynamic wheelchair motion revealed that vibration, rapid head movements, and complex backgrounds increased the probability of misrecognition and reduced system stability.

Although the proposed system demonstrated stable overall performance for embedded smart wheelchair navigation, several limitations remained. The system was still sensitive to low-light conditions, environmental disturbances, excessive dynamic motion, and highly complex backgrounds, while the observed response delay ranging from 0.784 seconds to 1.37 seconds indicates that real-time responsiveness was still constrained by image processing overhead and embedded hardware limitations.

Future work should therefore focus on improving environmental robustness and computational efficiency through adaptive background filtering, lightweight motion stabilization, motion prediction strategies, and hardware optimization using more powerful embedded platforms. Further validation involving more diverse participants, wider age ranges, different disability conditions, more complex indoor and outdoor environments, varied lighting conditions, and different wheelchair platforms is also necessary to improve the robustness, generalizability, and practical applicability of the proposed system. The integration of lightweight AI-based approaches may also improve recognition stability and responsiveness under more complex real-world operating conditions.

Overall, the proposed Adaptive Optical Flow approach demonstrates strong potential as a lightweight, low-cost, and non-invasive smart wheelchair control system for real-time embedded assistive technology applications.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Universitas Internasional Batam for providing research grants and facilities that enabled the successful completion of this study. This research is expected to contribute to improving the global reputation of Universitas Internasional Batam.

REFERENCES

[1] Pantelaki, E., Maggi, E., Crotti, D. (2021). Mobility impact and well-being in later life: A multidisciplinary systematic review. Research in Transportation Economics, 86: 100975. https://doi.org/10.1016/j.retrec.2020.100975

[2] Utaminingrum, F., Fauzi, M.A., Wihandika, R.C., Adinugroho, S., Kurniawan, T.A., Syauqy, D., Sari, Y. A., Adikara, P.P. (2017). Development of computer vision based obstacle detection and human tracking on smart wheelchair for disabled patient. In 2017 5th International Symposium on Computational and Business Intelligence (ISCBI), Dubai, United Arab Emirates, pp. 1-5. https://doi.org/10.1109/ISCBI.2017.8053533

[3] Herman, H., Kumar, Y.J., Wee, S.Y., Perhakaran, V.K. (2024). A systematic review on deep learning model in computer-aided diagnosis for anterior cruciate ligament injury. Current Medical Imaging Formerly Current Medical Imaging Reviews, 20(1): e15734056295157. https://doi.org/10.2174/0115734056295157240418043624

[4] Zhang, Z., Xu, P., Wu, C., Yu, H. (2024). Smart nursing wheelchairs: A new trend in assisted care and the future of multifunctional integration. Biomimetics, 9(8): 492. https://doi.org/10.3390/biomimetics9080492

[5] Wahyufitriyani, C., Susmartini, S., Priadythama, I. (2016). Review of intelligent wheelchair technology control development in the last 12 years. In 2016 2nd International Conference of Industrial, Mechanical, Electrical, and Chemical Engineering (ICIMECE), Yogyakarta, Indonesia, pp. 201-206. https://doi.org/10.1109/ICIMECE.2016.7910458

[6] WHO. (2023). Wheelchair provision guidelines. https://www.who.int/publications/i/item/9789240074521.

[7] Munotyaani, S., Pamosoaji, A.K., Bawono, B., Bawono, N.M.H., Sutrisno, Adi, A.N.M.P., Anggoro, P.W. (2025). Design of smart wheelchairs integrated with voice and joystick control, and obstacle detection system for Zimbabwe people: A review paper. Disability and Rehabilitation: Assistive Technology, 21(4): 1027-1050. https://doi.org/10.1080/17483107.2025.2596246

[8] Haddoun, A., Djabri, D., Saidani, M., Benbouzid, M.

(2025). Development and evaluation of a head-controlled wheelchair system for users with severe motor impairments. MethodsX, 15: 103485. https://doi.org/10.1016/j.mex.2025.103485

[9] Simpson, R.C. (2005). Smart wheelchairs: A literature review. Journal of Rehabilitation Research and Development, 42(4): 423-436. https://doi.org/10.1682/jrrd.2004.08.0101

[10] Dey, P., Hasan, M., Mostofa, S., Rana, A.I. (2019). Smart wheelchair integrating head gesture navigation. In 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, pp. 329-334. https://doi.org/10.1109/ICREST.2019.8644322

[11] Somawirata, I.K., Utaminingrum, F. (2023). Smart wheelchair controlled by head gesture based on vision. Journal of Physics: Conference Series, 2497: 012011. https://doi.org/10.1088/1742-6596/2497/1/012011

[12] Abdulkareem, H.J., Gharghan, S.K., Mutashar, S. (2025). Intelligent wheelchair control system using real-time head movement detection. Mathematical Modelling of Engineering Problems, 12(7): 2373-2386. https://doi.org/10.18280/mmep.120717

[13] Cui, J., Shang, Y., Yu, S., Wang, Y. (2024). Research on intelligent wheelchair multimode human-computer interaction and assisted driving technology. Actuators, 13(6): 230. https://doi.org/10.3390/act13060230

[14] Nalakurthi, N.V.S.R., Abimbola, I., Ahmed, T., Anton, I., Riaz, K., Ibrahim, Q., Banerjee, A., Tiwari, A., Gharbia, S. (2024). Challenges and opportunities in calibrating low-cost environmental sensors. Sensors, 24(11): 3650. https://doi.org/10.3390/s24113650

[15] Ammar, A., Ben Fredj, H., Souani, C. (2021). Accurate realtime motion estimation using optical flow on an embedded system. Electronics, 10(17): 2164. https://doi.org/10.3390/electronics10172164

[16] Shah, S.T.H., Xiang, X.Z. (2021). Traditional and modern strategies for optical flow: An investigation. SN Applied Sciences, 3(3): 289. https://doi.org/10.1007/s42452-021-04227-x

[17] Othman, R.B.M., Yuniarno, E.M. (2024). Smart wheelchair control based on spatial features of hand gesture. In 2024 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, pp. 1-6. https://doi.org/10.1109/CENIM64038.2024.10882756

[18] Nguyen, T., Ngo, B., Nguyen, T. (2025). Vision-based hand gesture recognition using a YOLOv8n model for the navigation of a smart wheelchair. Electronics, 14(4): 734. https://doi.org/10.3390/electronics14040734

[19] Ma'mUriyah, N., Yulianto, A., Li, L. (2024). Design prototype of audio guidance system for blind by using raspberry pi and fuzzy logic controller. Journal of Physics: Conference Series, 1230: 012024. https://doi.org/10.1088/1742-6596/1230/1/012024

[20] Chatterjee, S., Roy, S. (2021). A low-cost assistive wheelchair for handicapped & elderly people. Ain Shams Engineering Journal, 12(4): 3835-3841. https://doi.org/10.1016/j.asej.2021.04.021