


Realization of Gesture Control Application on Openmv Board Using Optical Flow in Real-Time Video Images

Bulent Turan 

Tokat Gaziosmanpasa University, Department of Computer Engineering, Tokat, Turkey

ABSTRACT

OpenMV Board is designed for purpose of non-complex image processing applications. It is an image processing sensor that has been a MicroPython embedded operating-system(OS).

In the study, it is aimed to develop gesture control applications for electrical household appliances and small budget devices. Therefore, the hardware to be used should be cheap and the algorithm should be simple. Thus three gesture control applications have been developed by using OpenMv board for use in different electrical appliances. These are 1-level control, 2-multi-component simple system control and 3-page flip. The algorithms used in the study are independent of the user because they are optical flow-based. Thus, the use of low-cost simple gesture control applications for industrial purposes (electrical appliances) can be realized.

Algorithms developed for applications were written on the OpenMV IDE. These application results were monitored in real-time through the IDE. In addition, the algorithm developed for level control has been embedded and tested on an SD card on OpenMv independent of OpenMV IDE. During the test, output information was generated using OpenMV pins and the level indicator created using yellow, green and red LEDs connected to the pins was checked real-time. Thus, the algorithm was tested on a computer-independent embedded system.

Keywords: Gesture control, Low-cost control with image processing, Simple gesture control applications, Image processing sensors, Open source SmartCam

Article History:

Received: 2020/12/01

Accepted: 2021/03/29

Online: 2021/06/30

Correspondence to: Bulent Turan,

E-mail: bulent.turan@gop.edu.tr;

Phone: +90 505 737 6766.

INTRODUCTION

Today, developments in the field of human-computer interaction (HCI) exponentially grows. HCI is most fundamentally carried out with keyboard and mouse. Humanity has been making efforts to develop the channel of interaction in this field. Today touch screens (especially indispensable on mobile devices), wearable devices, etc. are actively used in HCI. The technologies developed in this area and the studies that are still in progress can vary widely according to the purpose and place of use. For example, using webcam images to control of the cursor on the computer screen with the eye movements [1]. Control of smart TV, etc. systems [2, 3], control of humanoid robots [4], control with palm movement of the in-car multimedia device [5] can be realized by detecting and tracking of body movements with Kinect camera images.

Especially in the 2000s, studies have intensified. Due to the fact that gesture control concerns all parts of the body, special sensors can be used in the region where the studies are focused and the data obtained can be processed with different methods. For example, only data received with an accelerometer are processed using statistical analysis methods [7, 8], while images captured with cameras are processed using image processing methods [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Also, both data received using the PIR sensor and accelerometer were used together with image processing methods and statistical analyzes [20, 21]. Even, studies were to determine the gestures to be accepted by the users and have been conducted out to determine which gesture will be used for what purpose [7, 22]. In addition, there are studies to compare the studies in the literature. These studies also determine the advantages and secrets of the techniques used under different conditions [23, 24].

Studies in the field of HCI began in the 1980s [6].

Study;

- Within probability of use in small/daily household appliances,
- Low cost,
- It consists of gestures that each user can simply perform

They were designed to develop gesture control algorithms.

To this end, the gestures were first identified to enable the user requests to be digitized. In determining the gestures, similar studies [7,20] have been examined. It has been tried to choose the gestures that will be catchy and unforgettable for users. Determined gestures should be able to digitize user requests such as level control, multi-component system control, page flip application. It must also consist of hand movements that can be easily repeated by each user.

In order to lower the application cost, the source code (without licensing and programming problems) and circuit diagrams must be accessible by the public, capable of cost-effective image processing (only determination of the direction of movement is sufficient), and supported by the embedded OS. The preferred OpenMV Board is an image processing sensor with MicroPython embedded OS that meets all these requirements. Image processing sensors can be developed from simple to complex for very comprehensive image processing applications. This can lead to a very wide range of costs. However, the study focuses on the movement direction of the user's hand movements and the follow-up of the sequential movements. OpenMV board allows the cost of the hardware to remain at the desired levels.

A wide range of work in the field of gesture control leads to different levels of technology at the same time. On the one hand, high technology studies including machine learning, artificial intelligence, image processing techniques and 3D are continuing in the laboratory. These studies are used in developed and specific applications today. However, they are high-cost applications. On the other hand, a single image processing function for industrial purposes and software algorithms can be carried out. Especially in applications such as simple home appliances and in-car multimedia control, it is possible to achieve sufficient results with optics flow information in the moving image and optimized studies with a good software algorithm. In such studies, many gestures can be defined when a steady determination and follow-up of organs such as hands, fingers, body, and eyes can be made. Using these gestures, many control applications can be performed. Simple industrial applications can be controlled with simple image processing methods by identifying a small number of gestures. For example, if we only want to perform a control operation on a multimedia

system, it is sufficient to perform a few basic checks such as switching on/off, changing channels, increasing or decreasing the volume. Or, it is sufficient to perform the controls in the form of on-off, forward-back page-turning for the reading of an electronic book.

In this study, no organs of the user were detected and followed in the images captured from the camera. The aim of the study was to determine the direction of movement in a real-time image. With the algorithm developed using motion direction data, certain movements have been determined in a stable manner. Thus, the algorithm developed can be operated quickly on a card with a camera module and used for industrial purposes.

RELATED WORK

In the literature, it has been tried to develop gesture control applications by using different equipments. Some of those;

Image Processing Based Gesture Control Studies

Lee et al. have tried to develop a smart TV system with body movements. They used a Kinect camera in their study. Program opening, channel switching (increase/decrease), web page scrolling, return to homepage and exit are all expressed by separate body movements [2]. The necessity of using a special camera for this study is a disadvantage in terms of cost. Communication with the television can only be achieved by standing up and determined body movements. The challenge of standing up and the difficulty of carrying out wide-area movements is another disadvantage of this study.

Bellmore et al. have preferred Kinect camera in their work. They used to face and gesture control in their study [3]. It has similar disadvantages with other work with Kinect cameras. The method also requires a separate calibration for each user. The calibration stage can be unsuccessful according to the user's special situation (plenty of clothes, bags etc., having children, etc.) [3]. This is a major disadvantage because it can not be used in simple applications. Yavşan used a Kinect camera in his study, detected human upper body movements in real-time, and transferring them to a humanoid robot, simulating the human being by the robot [4]. A similar study was performed without using Kinect

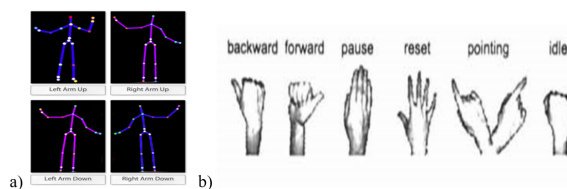


Figure 1. a) Kinect based 3D human gestures [5] b) Akyol et al. gestures and functions they use in their study [10]

camera by Kaura [15]. He captures images of the palm of the user at close range and morphologically determined the direction of motion and guided the robot [15]. On the other hand, Raghu produced control signals by detecting and following his hand similarly. He developed the application in a Matlab environment and did not use it as embedded [16].

Rahman et al. in their work, aimed to control the in-car multimedia device with hand movements. Due to the use of simple hand movements in the narrow space [5], there are many similarities with my study. However, in this study, the use of a Kinect camera will obviously affect the cost negatively. In addition, only one application was developed in the study [5]. The developed application in the study, is similar to the proposed multi-component simple system control algorithm in my study. But the other two algorithms do not meet. The sample images they use for the detection of movements with the Kinect camera was shown in Fig 1a.

In another study, an image processing-based gesture control algorithm was recommended. In this study, both the camera used and simple gestures in the narrow space was recommended for the control process [10]. For this reason, it is advantageous for simple systems. However, in order for the study to be carried out, the hand must be recognized and followed up [10]. Recognition and tracking increase the processing load in such applications. For this reason, it can be advantageous in real-time applications by reducing the processing load and removing the recognition and follow-up operations from the algorithms. The sample images they use for the detection of movements are shown in Fig 1b.

Accelerometer Based Gesture Control

Mantjarvi et al. planned the gesture control based on the accelerometer. In their study, they examined what different users prefer for different functions and the complexity of these movements [8]. In general, it was observed that the movements that the users prefer for different functions differ [8]. Generally;

On – off	->	push –pull
Next – previous	->	right-left
Increase – decrease	->	up – down

They stated that hand gestures on the right can be accepted to represent functions on the left. [8]. These gestures, which are determined for accelerometer-based studies, can be used in image processing-based applications.

Also, Kela and al. were worked to identify gestures for accelerometer-based VCR control [7]. Similarly, in this study next and previous were expressed in the same way with right and left movements by users. Up and down movements are adopted for play and stop. As a result, the acceptance of the

gestures may vary slightly depending on the characteristics of the device to be controlled

MATERIALS

These algorithms developed in the study will be run on the OpenMv board. OpenMV; It has a 216 MHz ARM Cortex M7 processor, 512 KB RAM, 64 KB heap space, MicroPython embedded operating system and a 0.3 MP camera which can be captured 640x480 sized grayscale image [25]. The IDE has been developed by the developer to program the OpenMv board. Applications can be developed with MicroPython embedded OS. In addition, the developer presents examples of card-specific image processing functions available to programmers.

Some examples of applications and functions offered by the developer are given below.

- Image filters
- Optical flow function
- Morphologic functions
- Edge functions
- Feature detection
- Face detection
- Eye tracking
- Color tracking
- Motor control applications
- Machine learning applications
- Barcode applications

In addition, applications and functions continue to be developed by the developer.

The study focuses on the movements in the image. Therefore, it is important to determine the optical flow as stable. The optical flow application on the OpenMV board is performed with the `find_displacement` function. The function is applied to two consecutive images and returns the changes in x-direction (Δ_x), y-direction (Δ_y), and response coefficient between the previous image and the next image. The fact that the response value is close to 1 indicates the high-reliability rate. It is accepted that there is no trouble as long as the response value is above 0.2

Table 1. Optical flow application (`find_displacement`) made in OpenMV compared with optical flow application (`estimateFlow`) made in MATLAB.

	<i>find_displacement</i> OpenMV	<i>estimateFlow</i> MATLAB
input	Consecutive two images of gray level	Consecutive two images of gray level
output	$\Delta_x, \Delta_y, Response$	$V_x, V_y, Orientation, Magnitude$

- Δ_x ; Gives information about the change in x-direction within the image
- Δ_y ; Gives information about the change in y-direction within the image.

- V_x ; X variable of change rate calculated separately for each pixel
- V_y ; Y variable of change rate calculated separately for each pixel

Similarly, the estimateFlow function gives information about the changes in x and y direction in the values obtained from MATLAB optical flow application. However, MATLAB's estimateFlow function gives these values based on pixels. This provides an advantage for some applications. For example, when there is more than one movement in the image, comprehensive studies can be done by determining regionally different movements. However, in studies where it is desired to determine a single movement, these values should be processed and updated to give a single output. This is an additional process load for the programmer.

- Response ; It is the coefficient of confidence for Delta_x and Delta_y
- Orientation ; Gives the phase angle of the optical flow in radians.
- Magnitude ; Express the amplitude of the optical flow

The response value calculated with the find_displacement function is produced for each image, while the orientation and magnitude values calculated with the estimateFlow function are produced separately for each pixel.

In this study, it has been performed the find_displacement function on OpenMV card is aimed to determine a single movement at the same time. Simple gesture control applications using the find_displacement function with OpenMV card can easily be performed.

"Hand movements planned to be used in order to control any simple device have been given in Fig. 2. It was foreseen that hand movements can be detected at a distance of 10-60 cm from the camera. In practice, it was not planned to develop image processing techniques to find hand or hand position within the image. Instead, the detection of the direction of movement and the determination of the directions of successive motion have been sufficient to carry out the control application. At this point, it can be assumed that movements (such as any object) will occur outside the hand

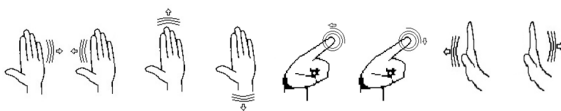


Figure 2. Respectively right-left, up-down, rotate clockwise- rotate counterclockwise and push-pull hand gestures

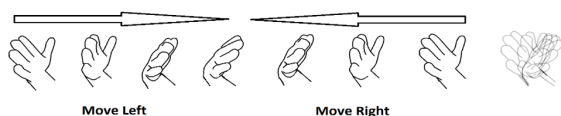


Figure 3. Page turning gesture

movement in the area of interest (ROI) of the camera and that the control operation will be unstable. In order to solve this problem in the study, it was planned to activate the control system with a combination of sequential movement. Also In addition, the control system has been deactivated when no significant movement was taken for a certain period of time.

METHOD

Three different algorithms and one application have been developed on OpenMV card. These;

- Level control algorithm,
- Multi-component Simple System Control algorithm,
- Page turning algorithm.
- Level control application

These controls can be performed using many different methods. Electronic based control systems that detect and operate the on-off switch time, systems that perceive the pressure force as an analog and adjust the level according to the magnitude of the force, level controls using a potentiometer, can be done with electromechanical, electronic or even computerized control methods.

Today, these controls can be performed using image processing methods. However, it takes time for the comprehensive control applications realized by image processing to be economically achievable. From this point of view, the applications developed out within the study aim to bring the advantages of image processing methods to the users without increasing the cost even in the simplest and cheapest electrical/electronic appliances.

In industrial R & D studies, simple, result-oriented, stable and at the same time low-cost solutions are preferred for the production stage. Thus, companies develop themselves in terms of competitiveness. High-cost solutions are therefore not preferred as much as possible. However, in case of obligation, also high-cost solutions are preferred.

Level Control Algorithm

Level control has been using often in many places in the industry and in our daily lives. For example, level control has been carried out in many applications in our daily life such as the voice of a multimedia device, opening level of the vehicle glass, light intensity level.

In the study, in order to develop an image processing based level control algorithm, it has been decided to use the circular gesture of the index finger. Movement of the index finger rotate clockwise and rotate counterclockwise was used for level control. By turning the index finger rotate clockwise, the level was increased and the level was reduced

ced by moving the index finger in the opposite direction. Although the level of the maximum level in the systems to be controlled varies, the level is expressed as a percentage in most systems. In other words, the ratio is expressed as corresponding to the %ratio. Due to the general use in this direction, the level of the study was decided to be in the range of 0-100. Therefore, the level value was limited to 0-100 in order to be compatible with many real-life applications.

Of the application developed, for realization on OpenMv Cam, the algorithm given below has been followed.

Level Control Algorithm

- Delta_x ve Delta_y is detected,
- Delta_x and Delta_y's polarization (Positive or Negative) is detected and saved,
- When the polarization of either Delta_x or Delta_y changes, the new polarization of both is determined and the last four polarizations are saved,
- Compare the last four polarizations of Delta_x and Delta_y,

$$\left(\begin{array}{l} \text{CONTROL LEVEL is increased} \\ \text{if appropriate to the combination} \\ \text{given below} \\ \begin{pmatrix} \Delta x & \Delta y \\ + & - \\ + & + \\ - & + \\ - & - \end{pmatrix} \end{array} \right) \quad \left(\begin{array}{l} \text{CONTROL LEVEL is reduced} \\ \text{if appropriate to the combination} \\ \text{given below} \\ \begin{pmatrix} \Delta x & \Delta y \\ + & + \\ + & - \\ - & - \\ - & + \end{pmatrix} \end{array} \right)$$

The results obtained by moving the index finger clockwise and counter-clockwise using the software prepared in the OpenMV IDE were given in Fig. 4.

Fig 4 will be seen when examined, the clockwise movement of the index finger increases the level value. The counter-clockwise movement of the index finger decreases the level value. The level is called "CONTROL LEVEL" in the

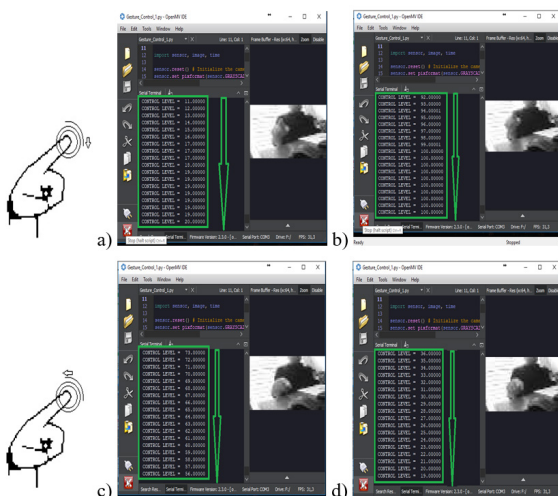


Figure 4. a-b Clockwise rotation c-d counter-clockwise rotation gesture and resulting output values

developed software and algorithm.

The level is set in the range 0-100 and the level resolution in this range is 1 unit, it takes some time to adjust the level values in this long-range. In addition, it is necessary to repeat the sequential rotational movements in the same direction. Although random movements in the sensor visual field may cause changes in the level value, it is not possible to have a change in the continuous same direction as a result of random movements. Also, the average change in random movements will be zero. For this reason, no key movement combination has been planned for the use phase of the developed algorithm.

Multi-Component Simple System Control Algorithm

Many products for the end-user (home appliances etc.) have more than one component to be controlled.

For example, the following devices, etc. which we use in daily life, include multi-component systems;

- Paddle box; engine level and lighting level,
- Microwave oven; program selection and duration,
- Electrical oven; program selection, heat and time,
- Multimedia systems (radio, cd player, aux); source selection channel selection and volume adjustment,

In these systems, is first determine which component to control, then we adjust the level of the component. Therefore, in this study, it is necessary to determine the gestures that produces at least two different results. For the realization of this selection was determined right-left, up-down hand movements. With up-down palm movements the desired component was selected, the level of the component was controlled with another gesture (right-left palm movements).

In the study, it was preferred to use the up-down and right-left gestures in front of the camera of the palm for multi-component system control. In multi-component system control, the level limits of the components may vary. At the same time, the number of components may vary. In the study, the algorithm was developed for two components consisting of three levels. If necessary, the number of components and the operating range of the components can be updated via an algorithm.

Negative objects and gestures incoming to ROI should not adversely affect to component selection and levels. For this, a lock of gesture combinations has been developed. Component selection and level control cannot be performed, if it does not occur the gesture combinations. However, after a certain time the control operation has been completed the lock of gesture combinations was automatically activated.

The lock combination was carried out by the user moving of the palm to form a square up-right-down-left in succession against the camera. This lock combination was controlled in the second step of the algorithm.

Of the application developed, for realization on OpenMv Cam, the algorithm given below has been followed.

Multi-Component Control Algorithm

- Delta_x ve Delta_y is detected,
- Combination of lock gesture took place -> is control active. Or no -> is control pasive
- Lock timeout is checked

```

( If |delta_y| > |delta_x| and delta_y < 0 => number1 = number1 + 1
  If number1 >= 20 => control1 = 1, number1 = 0
  If |delta_x| > |delta_y| and delta_x > 0 and control1 = 1 => number2 = number2 + 1
  If number2 >= 20 => control2 = 1, number2 = 0
  If |delta_y| > |delta_x| and delta_y > 0 and control2 = 1 => number3 = number3 + 1
  If number3 >= 20 => control3 = 1, number3 = 0
  If |delta_x| > |delta_y| and delta_x < 0 and control3 = 1 => number4 = number4 + 1
  If number4 >= 20 => control4 = 1, number4 = 0
)

```

- If there is movement in ROI when the lock is active,
 - (If 150 cycles (approx. 5 sec) no gesture (or any motion) is detected, the lock is disabled.
 - (If gesture (or any motion) is detected, the lock remains active and the number of cycles is reset.
- the direction of movement is determined.

```

( If |delta_x| > |delta_y| and delta_x > 0 and control4 = 1 => direction of gesture RIGHT
  If |delta_x| > |delta_y| and delta_x < 0 and control4 = 1 => direction of gesture LEFT
  If |delta_x| < |delta_y| and delta_y > 0 and control4 = 1 => direction of gesture DOWN
  If |delta_x| < |delta_y| and delta_y < 0 and control4 = 1 => direction of gesture UP
)

```

- > direction of gesture UP => CONTROL ACTIVE level is increased (0,1,2,3)
- > direction of gesture DOWN => CONTROL ACTIVE level is reduced (3,2,1,0)
- > direction of gesture RIGHT and CONTROL ACTIVE level 0 => no action is taken
- > direction of gesture RIGHT and CONTROL ACTIVE level 1 => The motor level is increased (0,1,2,3)
- > direction of gesture RIGHT and CONTROL ACTIVE level 2 => The light level is increased (0,1,2,3)
- > direction of gesture LEFT and CONTROL ACTIVE level 0 => no action is taken
- > direction of gesture LEFT and CONTROL ACTIVE level 1 => The motor level is reduced (3,2,1,0)
- > direction of gesture LEFT and CONTROL ACTIVE level 2 => The light level is reduced (3,2,1,0)

Multi-Component Control Algorithm developed by following the algorithm given below was run real-time on OpenMV IDE by connecting with OpenMV. Outputs obtained during the study are given in Fig. 5.

In practice, 2 components are controlled with CONTROL ACTIVE. When CONTROL ACTIVE is 0, no compo-

nent is controlled. When the CONTROL ACTIVE is 1, the Motor Level operation is checked. The Motor Level has 4 levels of operation. If Motor Level = 0, the motor does not work, if Motor Level = 1-3, motor operation level sets.

If CONTROL ACTIVE = 2, Light Level operation is checked. The Light Level has 3 levels of operation. If Light Level = 0, Light does not work, if Light Level = 1-2, Light operation level sets.

As can be seen from Fig 5, the UP-DOWN movement of the palm allows us to select the component to be controlled. (CONTROL ACTIVATE = 0-2). The RIGHT-LEFT movement of the palm allows the value of the component to be increased or decreased. (Engine Level = 0-1-2-3).

Page Turning Algorithm

In many virtual book applications such as e-books, e-catalogs, is performed page turn process using any of the physical contact methods (touch, clicks etc.). Similarly, in many digital applications, up-down, right-to-left shifts are performed using physical contact methods. Whereas these processes can also be physical contactless with OpenMv Cam. In this application, page-turning and page reversal turning was performed with image processing-based gesture control.

In fact, the page-turning process and page reversal turning differ in real-life. However, when this process is considered as the turn of the pages of a book in a digital environment with the image processing, the gesture of the hand in one direction can be regarded as turning the page in this direction, and the reverse gesture of the hand direction can be considered as the page reversal turning. In this dilemma, which is logically correct, it may be problematic if the gestures are performed consecutively. For example, the page-turning gesture must be repeated 10 times for the book to turn 10 pages. However, this gesture will be done 10 times in the forward direction and 9 times in the reverse direction, and in fact only 1 page-turning will be performed.

Elimination of this problem, which direction we want to turn the page (forward-back) depends on our ability to transfer to the smart camera. Push gesture has been used to transfer of this data to the smart camera. The direction of rotation turning of the page is determined by the pushing gesture. Also with page-turning gesture it is turning at the desired number. Each push action changes the current position (page-turning/page reversal turning). Thus, page-turning and reverse page-turning gestures are prevented from interfering with each other.

Of the application developed, for realization on OpenMv Cam, the algorithm given below has been followed.



Figure 5. a-b Up and down c-d right and left movements of the palm and output values obtained

Page Turning Algorithm

- Delta_x ve Delta_y is detected,
- Page turning direction determined,
 - $reverse = 0 \Rightarrow forward$ / $reverse = 1 \Rightarrow backward$
 - Check the push gesture
 - (If $delta_x < 0$ and $delta_y > 0 \Rightarrow condition = condition + 1$)
 - push gesture occurred \Rightarrow change page turning direction
 - (If $condition \geq 25$ and $reverse = 1 \Rightarrow reverse = 0, condition = 0$)
 - (If $condition \geq 25$ and $reverse = 0 \Rightarrow reverse = 1, condition = 0$)
- The page turning gesture is detected and the page is turned forward if the turn direction is forward
 - (If $|delta_x| > |delta_y|$ and $delta_x < 0$ and $reverse = 0 \Rightarrow number_1 = number_1 + 1$)
 - If $number_1 \geq 5 \Rightarrow forward$ page turning, $number_1 = 0$
- The page turning gesture is detected and the page is turned backward if the turn direction is backward
 - (If $|delta_x| > |delta_y|$ and $delta_x > 0$ and $reverse = 1 \Rightarrow number_2 = number_2 + 1$)
 - If $number_2 \geq 5 \Rightarrow backward$ page turning, $number_2 = 0$

No need for any lock combinations for this algorithm. Because if the user is doing right or left random movements in front of the camera, it is not already focused on the page. This is not a problem for the operation of the system. The software developed by following the given algorithm was run on OpenMV IDE by connecting with OpenMV. Outputs obtained during the real-time operation were given in Fig. 6.

When Fig. 6 are examined, it has been seen that the algorithm changes the detection direction after each push motion. For example, when a pushing gesture was performed while sensing page turn movement in the forward direction, the software stops detecting the page turn movement in the forward direction and begins to detect the page turn movement in the reverse direction.

Level Control Application

The program written with the algorithm developed for level control has been embedded in the OpenMV image sensor. Thus, the suitability to work on the embedded

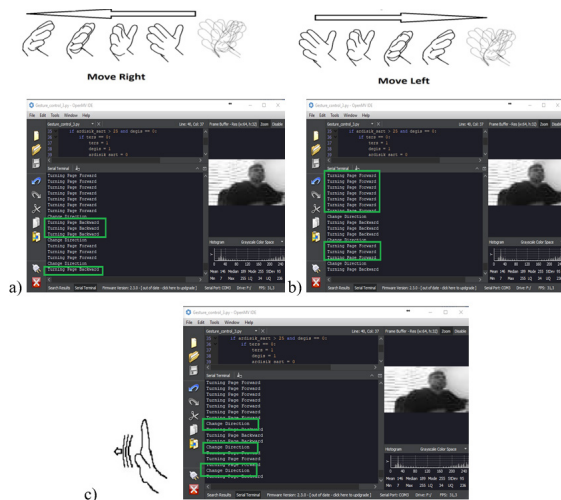


Figure 6. a) Right page b) left page-turning gesture and resulting output values c) Push gesture and output values obtained

system has been tested. There is actually no difference between running the software on the OpenMV card and working on the OpenMV IDE. Because the OpenMV IDE was developed to match the capabilities of OpenMV Cam. Hence, the software is performing the same performance on the IDE and on the card.

OpenMV Cam has serial and parallel communication pins (I/O) for communication with other hardware (electronic cards, sensors, indicators, etc.). In the developed application, parallel communication was preferred due to the fact that the number of pins is sufficient and the possibility of direct supply (no additional equipment is needed). However, UART, I2C, SPI serial communication pins can also be used in studies where the number of pins is not sufficient or more comprehensive studies.

The program prepared by following the Level Control Algorithm was applied to the level indicator by using 10 steps led (4 green-3 yellow-3 red). Normally, the program outputs 0-100 level and 1 unit resolution. However, since the use of 100 indicator leds would not be very efficient during the application, the data resolution transferred to the external environment was reduced by 10 times. Only 10 different levels have been transferred to external environment using OpenMV pins. The indicator control was made using the pins on the OpenMv. In the table 2 are the OpenMv I/O and

Table 2. OpenMv I/O and control level

OpenMv I/O	Control Level	Led Level
Pin0	0-10	1 Green
Pin1	10-20	2 Green
Pin2	20-30	3 Green
Pin3	30-40	4 Green
Pin4	40-50	5 Yellow
Pin5	50-60	6 Yellow
Pin6	60-70	7 Yellow
Pin7	70-80	8 Red
Pin8	80-90	9 Red
Pin9	90-100	10 Red

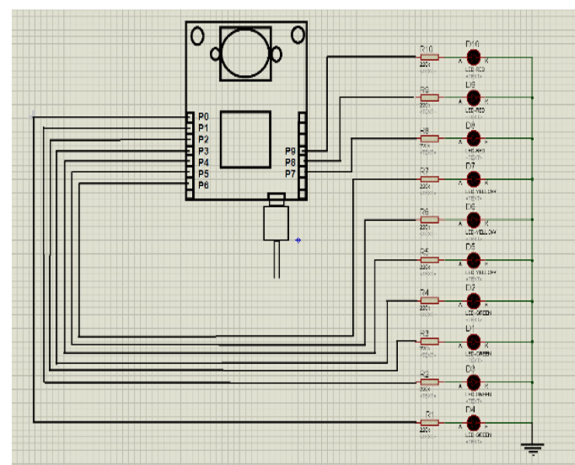


Figure 7. Level control application circuit diagram

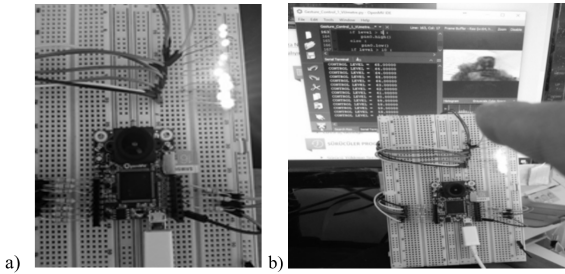


Figure 8. Level control application images

indicator levels. The circuit diagram of the developed level control application was given in Fig. 7 and output images were given in Fig. 8.

RESULTS

In this study, image processing-based gesture control algorithms has been developed to be used in low-cost embedded systems. The cost of such embedded systems affects hardware and software together. Since the scope of the software to be developed is wide or includes advanced image processing methods, it will have a negative effect on the cost as it will increase the hardware requirement. Therefore, the software to be developed should be able to run on small budget hardware. In this study OpenMV Cam was preferred as hardware. The algorithms used in the study are based on the find_displacement function that can be implemented on OpenMV Cam. Thus, gesture control applications can be realized with low-budget hardware without using advanced image processing methods.

OpenMv board was used to develop gesture control applications. Applications developed using the researcher's developed functions, on the OpenMv IDE have worked stably. The study has been used the find_displacement function based which developed by the researcher. This function provides the determination of the changes occurring between consecutive images. Thus, the researchers can develop a gesture control application by following the direction of movements in the image. Developed as a raw image sensor, OpenMv can be used as an image sensor by the users/researchers. MicroPython embedded OS based functions were available for users to develop applications on OpenMV.

The level control, multi-components simple system

Table 3. Test results

	Level Control Algorithm		Multi-Component Control Algorithm		Page Turning Algorithm	
	<i>inexperienced</i>	<i>experienced</i>	<i>inexperienced</i>	<i>experienced</i>	<i>inexperienced</i>	<i>experienced</i>
<i>Person1</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>Person2</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>
<i>Person3</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>
<i>Person4</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>
<i>Person5</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>Total</i>	<i>60%</i>	<i>100%</i>	<i>40%</i>	<i>100%</i>	<i>%40</i>	<i>100%</i>

control and page-turning algorithms developed in the study, can be embedded on the OpenMV board and can be used to control any multimedia device or any household electrical appliance. OpenMV source code and schematic drawings are open to help reduce costs. It contributes to a positive increase in cost-performance ratio. Thus, the OpenMV image sensor can be used as hardware in low-cost industrial applications. However, the software also to be used together with the hardware should not contain high technology. Otherwise, the cost will adversely affect the performance ratio. At the same time, high-tech software on low-cost hardware cannot already be implemented. For example, operations with the Kinect camera cannot be performed on OpenMV Cam.

In this study, the software based on the developed algorithms were not developed specifically for a particular user. So it can be used by all users. In addition, they do not require calibration at every restart as in some applications. However, the system response may not be the same for every user. Because each user's preferred movement speed and styles vary. According to the way users perform gestures, the system can make more mistakes for some users. However, although the users tend to make different choices, the necessary corrections for the control will occur spontaneously after certain usage.

In the study "Optical flow-based gesture motion direction recognition method", up-down and right-left directions are determined. It has been stated that it is sufficient to detect movements in the same direction twice for these determinations. The algorithm has been developed accordingly. It is like the Multi-Component Simple System algorithm in my study. Success rates in the study are not given in numerical values. In both studies, the system is independent of the user. At the same time, the user's success in using the system is related to the usage experience. The most important difference between the two studies is the wide range of level control (Level Control Algorithm) and the page turning control (Page-Turning Algorithm). These controls are not carried out in the study [26].

Pathak and Jalal [27] determined eight different hand movement directions (right-left, up-down and intermediate directions) in their studies. With different combinations of these directions of movement, they identified the words

in the Indian sign language. With the method they developed, they first used 15% of the recorded images to create the sequential code combination and the remaining 85% for testing. As a result, they achieved 90.4% success. The focus on sign language of the study is the difference. They expressed the recurring direction determinations as residuals and filtered them. Because repeatedly repeated direction determinations cause incorrect code combinations [27]. In the method proposed in this study, repetitive direction determinations express the level change and are not filtered. The method suggested in the study is independent of the user, as it focuses on the direction of movement as in this study.

Hand movements should not be too fast or too slow in order to be detected correctly and to ensure stable operation of the algorithms. In general, it is thought that the user experience will experience the stable operation of the algorithms and calibrate the movement speed itself. At the same time, during the application phase, the movement speeds were approximately determined as follows.

- Clockwise or counter-clockwise rotation speed is 1 turn/sec for the Level Control Algorithm.
- Up-down or right-left movements speed is 0.5 level/sec for Multi-Component Control Algorithm.
- Right or left page-turning gestures speed is 1 sec for Page Turning Algorithm.

After the developed algorithms were completed, they were tested by 5 person who used the system for the first time. Test results are given below.

After the users were informed, they were enabled to use the applications. In the Table 3, the first use results and the results after experiencing the applications for a certain period of time are given. In the first use, Level control, Multi-component control and page-turning algorithms were realized with 60%, 40% and 40% success, respectively. However, after the users experienced the applications, 100% success rate was achieved for all applications.

CONCLUSION AND RECOMMENDATIONS

- Developed algorithms can be run independently from the computer when embedded on OpenMV. Therefore, it is suitable for embedded system design.

- In the development phase, algorithms were tested by different users via OpenMV IDE and no differences were determined. In fact, the study has been based only on the determination of the direction of movement. Therefore, it does not have the ability to distinguish users.

- It was anticipated that the movements will be performed with an average speed, the number of repetitions in the

successive data is determined according to this prediction. For this reason, the rate of producing false results is increased in very fast and very slow-motion combinations. However, due to the fact that the machines to be controlled are machines such as small appliances etc, the erroneous results will not be a big problem and the users will adjust their movement speed according to the embedded system as in all technological devices.

The study was carried out on the OpenMV board. Special hardware can be developed for the developed algorithms. Also, applications are independent of the user but dependent on movement speed. The stable operating range can be expanded by developing filters for moving speed.

CONFLICT OF INTEREST

Authors approve that to the best of their knowledge, there is not any conflict of interest or common interest with an institution/organization or a person that may affect the review process of the paper.

References

1. Turan B, Eskikurt Hİ, Can MS. An application based on artificial neural network for determining viewpoint coordinates on a screen. *Elektronika Ir Elektrotechnika* 22 2 (2016) 86-91. DOI: <http://dx.doi.org/10.5755/j01.eie.22.2.7586>
2. Lee WPO, Kaoli C, Huang JY. A smart TV system with body-gesture control, tag-based rating and context-aware recommendation. *Knowledge-Based Systems* 56 (2014) 167-178. <https://doi.org/10.1016/j.knosys.2013.11.007>
3. Bellmore C, Ptucha R, Savakis A. Interactive display using depth and RGB sensors for face and gesture control. *Western New York Image Processing Workshop*, 2011. <https://doi.org/10.1109/WNYIPW.2011.6122883>.
4. Yavşan E, Uçar A. Gesture imitation and recognition using Kinect sensor and extreme learning machines. *Measurement* 94 (2016) 852-861.
5. Rahman ASMM, Saboune J, Saddik AE. Motion-path based in car gesture control of the multimedia devices. *DIVANet '11 Proceedings of the first ACM international symposium on Design and analysis of intelligent vehicular networks and applications* 69-76, 2016. <https://doi.org/10.1145/2069000.2069013>.
6. Bhuiyan M, Picking R. Gesture-controlled user interfaces, what have we done and what's next?. 2009. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.562.6140&rep=rep1&type=pdf>. (accessed 13 11 2018).
7. Kela J, Korpipaa P, Mantyarvi J, Kallio S, Savino G, Jozzo L, Marca S. Accelerometer-based gesture control for a desing environment. *Personal and Ubiquitous Computing* (2006) 285-299. <https://doi.org/10.1007/s00779-005-0033-8>
8. Mantyarvi J, Kela J, Korpipaa P, Kallio S. Enabling fast and effortless customisation in accelerometer based gesture

- interaction. MUM '04 Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia 25-31, 2004. <https://doi.org/10.1145/1052380.1052385>
9. Hackenberg G, McCall R, Broll W. Lightweight Palm and Finger Tracking for Real-Time 3D Gesture Control. 2011 IEEE Virtual Reality Conference, 2011. <https://doi.org/10.1109/VR.2011.5759431>.
 10. Akyol S, Canzler U, Bengler K, Hahn W. Gesture control for use in automobiles. IAPR Workshop on Machine Vision Applications, Nov. 28-30, 2000. The University of Tokyo, Japan. <https://pdfs.semanticscholar.org/fb51/6222c7c87f42872a28ff8fc74139447b1280.pdf>. (accessed 13 11 2018).
 11. Bizzotto N, Costanza A, Bizzotto L, Revis D, Sandri A, Mangan B. Leap motion gesture control with osirix in the operating room to control imaging. Surgical Innovation, 2014. <https://doi.org/10.1177/1553350614528384>.
 12. Cohen CJ, Beach G, Foulk G. A Basic Hand Gesture Control System for PC Applications. IEEE Xplore Digital Library, 2001. <https://doi.org/10.1109/AIPR.2001.991206>.
 13. Gallo L, Placitelli AP, Ciampi M. Controller-free exploration of medical image data: Experiencing the Kinect. 2011 24th International Symposium on Computer-Based Medical Systems (CBMS), 2011. <https://doi.org/10.1109/CBMS.2011.5999138>.
 14. Doğan RÖ, Doğan H, Köse C. Virtual Mouse Control with Hand Gesture Information Extraction and Tracking. 23rd Signal Processing and Communications Applications Conference (SIU) 2015. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7130228>. (accessed 10 12 2019).
 15. Kaura HK, Honrao V, Patil S, Shetty P. Gesture Controlled Robot using Image Processing. (IJARA) International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No. 5, 201, 2013 <https://pdfs.semanticscholar.org/ff0f/20e3dbbdf257ec3ca36be4ed251036b49e11.pdf>. (accessed 13 11 2018).
 16. Chowdary PRV, Babu MN, Subbareddy TV, Reddy BM, Elamaran V. Image Processing Algorithms for Gesture Recognition using MATLAB. 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2014. <https://doi.org/10.1109/ICACCCT.2014.7019356>.
 17. Kar S, Banerjee S, Jana A, Kundu D, Chatterjee D, Ghosh S, Mitra D, Gupta SD. Image Processing Based Customized Image Editor and Gesture Controlled Embedded Robot Coupled with Voice Control Features. (IJACSA) International Journal of Advanced Computer Science and Applications 6 (2015) 11. <https://pdfs.semanticscholar.org/f000/be3e91d69dc8ce1f87ae32ae7e5395b09b86.pdf>. (accessed 13 11 2018).
 18. Osimani C, Piedra-Fernandez JA, Ojeda-Castelo JJ, Iribarne L. Hand Posture Recognition with Standard Webcam for Natural Interaction, WorldCIST 2017. Advances in Intelligent Systems and Computing, vol 570 (2017) Springer.
 19. Hsiang-Yueh L, Hao-Yuan K, Yu-Chun H. Real-time Hand Gesture Recognition System and Application. Sensors and Materials, Vol. 30, No. 4 (2018) 869-884.
 20. Schacher JC. Gesture control is sounds in 3D space. Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07), New York, NY, USA, 2007. http://www.nime.org/proceedings/2007/nime2007_358.pdf. (accessed 13 11 2018).
 21. Erden F. Hand gesture recognition using two differential PIR sensors and a camera. 2014 22nd Signal Processing and Communications Applications Conference (SIU), 2014. <https://doi.org/10.1109/SIU.2014.6830237>.
 22. Şahin A. Hacking the Gestures of Past for Future Interactions. M.Sc. THESIS, 2013 <http://muep.mau.se/bitstream/handle/2043/15700/Atilim%20Sahin%20-%20Hacking%20the%20Gestures%20of%20Past%20for%20Future%20Interactions.pdf?sequence=2&isAllowed=y>. (accessed 13 11 2018).
 23. Rautaray S, Agrawal A. Vision based hand gesture recognition for human computer interaction: a survey. Artif. Intell. Rev. 43 (2015) 1-54
 24. Munir O, Ali AN, Javaan C. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. Journal of Imaging 6(8) 2020 73.
 25. OpenMv Cam M7 Specifications, (2017). <https://openmv.io/products/openmv-cam-m7>. (accessed 14 11 2018).
 26. Optical flow-based gesture motion direction recognition method, <https://patents.google.com/patent/CN104331151A/en>
 27. Pathak B, Jalal AS. Motion Direction Code-A Novel Feature for Hand Gesture Recognition. Advances in Intelligent Systems and Computing, vol 798 2019. Springer