A MINI PROJECT REPORT
On

Hand Gesture Recognition for Human Computer Interaction

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Course

Human Machine Interaction

In Computer Engineering (VIII SEM)

Submitted By

Aditya Shinde
Jitendra Phull
Rahul Wasnik

Subject Incharge
Sandhya Awate

Department Of Computer Engineering
A.C. PATIL COLLEGE OF ENGINEERING Kharghar – 410 210
UNIVERSITY OF MUMBAI

Academic Year 2019 – 20

Department of Computer Engineering
A. C. Patil College of Engineering Kharghar – 410 210

CERTIFICATE

This is to certify that the requirements for the project report entitled '**Hand Gesture Recognition for Human Computer Interaction**' have been successfully completed by the following students:

Name	Roll No.
Aditya Shinde	58
Jitendra Phull	50
Rahul Wasnik	68

in partial fulfillment of the course Human Machine Interaction in Computer Engineering (VIII SEM) of Mumbai University in the Department of Computer Engineering, A. C. Patil College of Engineering, Kharghar – 410 210 during the Academic Year 2019 – 20.

(Sandhya Awate)
Subject Incharge

DECLARATION

We declare that this written submission for Natural Language Processing mini project entitled “Project Title” represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not been taken when needed.

Project Group
Members:

Aditya Shinde
Jitendra Phull
Rahul Wasnik

Date:
Place:

Abstract

The use of a physical controller like mouse, keyboard for human computer interaction hinders natural interface as there is a strong barrier between the user and computer. In this paper, we have designed a robust marker- less hand gesture recognition system which can efficiently track both static and dynamic hand gestures. Our system translates the detected gesture into actions such as opening websites and launching applications like VLC Player and PowerPoint. The dynamic gesture is used to shuffle through the slides in presentation. Our results show that an intuitive HCI can be achieved with minimum hardware requirements.

1. Introduction

The basic goal of Human Computer Interaction is to improve the interaction between users and computers by making the computer more receptive to user needs. Human Computer Interaction with a personal computer today is not just limited to keyboard and mouse interaction. Interaction between humans comes from different sensory modes like gesture, speech, facial and body expressions. Being able to interact with the system naturally is becoming ever more important in many fields of Human Computer Interaction.

Both non-vision and vision based approaches have been used to achieve hand gesture recognition. An example of a non-vision based approach is the detection of finger movement with a pair of wired gloves. In general vision based approaches are more natural as they require no hand devices. Theoretically the literature classifies hand gestures into two types static and dynamic gestures. Static hand gestures can be defined as the gestures where the position and orientation of hand in space does not change for an amount of time. If there are any changes within the given time, the gestures are called dynamic gestures. Dynamic hand gestures include gestures like waving of hand while static hand gestures include joining the thumb and the forefinger to form the “Ok” symbol.

2. Related work

The literature survey conducted provides an insight into the different methods that can be adopted and implemented to achieve hand gesture recognition. It also helps in understanding the advantages and disadvantages associated with the various techniques. The literature survey is divided into two main phases i.e. the camera module and the detection module. The camera module identifies the different cameras and markers that can be used. The detection module deals with the pre-processing of image and feature extraction.

The commonly used methods of capturing input from the user that has been observed are data gloves, hand belts and cameras. The approach of gesture recognition [1] and [2] uses input extraction through data gloves. A hand belt with gyroscope, accelerometer and a Bluetooth was deployed to read hand movements are used [3] [4]. The authors [5] used a creative Senz3D Camera to capture both colour and depth information and [6] used a Bumblebee2 stereo camera. A monocular camera was used by [7]. Cost efficient models like [8], [9] and [10] have implemented their systems using simple web cameras. The methods [11] [12] make use of a kinect depth RGB camera which was used to capture colour stream. As depth cameras provide additional depth information for each pixel (depth images) at frame rate along with the traditional images [13] [14]. Most technologies allow a hand region to be extracted robustly by utilizing the colour space. These do not fully solve the background problem. This background problem was resolved in [15] by using a black and white pattern of augmented reality markers (monochrome glove). While inbuilt webcams do not give depth information, they require less computing costs. Hence in our model, we used a webcam available in the laptop without the use of any additional cameras or hand markers such as gloves.

A large number of methods have been utilized for pre -processing the image which includes algorithms and techniques for noise removal, edge detection, smoothening followed by different segmentation techniques for boundary extraction i.e. separating the foreground from the background. The authors [9] [16] used a morphology algorithm that performs image erosion and image dilation to eliminate noise. Gaussian filter was used to smoothen the contours after binarization [10] [17]. To perform segmentation, in [6] a depth map was calculated by matching the left and right images with the SAD (Sum of Absolute Differences) algorithm. In [6], the Theo Pavildis Algorithm which visits only the boundary pixels was used to find the contours. This method brings down the computational costs. In [9] [13] 16] the biggest contour was chosen as the contour of the hand palm after which the contour was simplified using polygonal approximation. Classification is a process in which individual items are grouped based on the similarity between the items. The approach [18] uses Euclidean distance based classifier to

recognise 25 hand postures. Support Vector Machine (SVM) classifier was used in [19] and [11]. We deviate from other traditional methods without using any hand markers such as gloves for gesture recognition. In our model, we used a webcam available in the laptop without the use of any additional cameras by making the system cost effective. Thus our system finds applications in day to day system.

3. Proposed Hand Gesture Recognition System

The overall system consists of two parts, back end and front-end. The back end system consists of three modules: Camera module, Detection module and Interface module as shown in Fig. 1. They are summarized as follows:

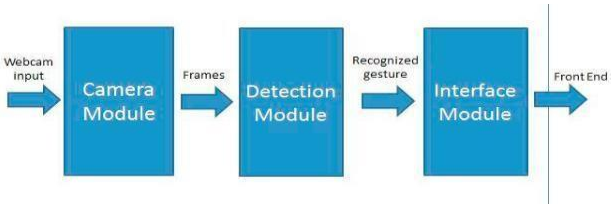

Fig. 1 Back end architecture

3.1 Camera module

This module is responsible for connecting and capturing input through the different types of image detectors and sends this image to the detection module for processing in the form of frames. The commonly used methods of capturing input are data gloves, hand belts and cameras. In our system, we use the webcam inbuilt which is cost efficient to recognize both static and dynamic gestures. The system has suitable provision to allow input from a USB based webcam as well but this would require some expenditure from the user. The image frames obtained are in the form of a video.

3.2 Detection module

This module is responsible for the image processing. The output from camera module is subjected to different image processing techniques such as colour conversion, noise removal, thresholding following which the image undergoes contour extraction. If the image contains defects, then convexity defects are found according to which the gesture is detected. If there are no defects, then the image is classified using Haar cascade to detect the gesture.

In the case of dynamic gestures, the detection module does the following; If Microsoft PowerPoint has been launched with a slideshow being enabled and the webcam detects palm in movement, for 5 continuous frames then the dynamic gesture swipe is detected.

3.3 Interface module

This module is responsible for mapping the detected hand gestures to their associated actions. These actions are then passed to the appropriate application. The front end consists of three windows. The first window consists of the video input that is captured from the camera with the corresponding name of the gesture detected. The second window displays the contours found within the input images. The third window displays the smooth thresholded version of the image. The advantage of adding the threshold and contour window as a part of the Graphical User Interface is to make the user aware of the background inconsistencies that would affect the input to the system and thus they can adjust their laptop or desktop web camera in order to avoid them. This would result in better performance.

4. Proposed method

We propose a marker less gesture recognition system,that follows a very efficient methodology as shown in fig. 2.

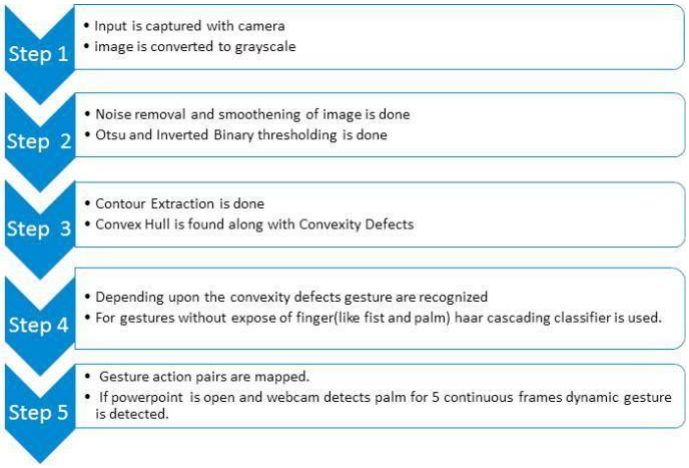

Fig. 2 Proposed method for our gesture recognition system

4.1. Noise removal and Image smoothening

The input image, which is in RGB color space, is cropped to a size of 300 * 300 pixels. It is then converted into a gray scale image. This process is shown in Fig. 3.

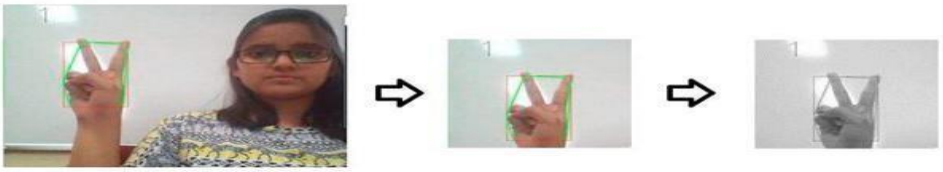

Fig. 3 Process of cropping and converting RGB input image to grey scale

Noise in images can be defined as a random variation of brightness or colour information that is usually produced during the image acquisition process from the webcam. This noise is an undesirable aspect of the image and needs to be removed. In order to do this, Gaussian filter is applied. Gaussian filtering is performed by the convolution of Gaussian kernel with each point in the input array. These are then added to produce the output array. A 2D Gaussian kernel can be represented mathematically as shown in Eqn. 1.

$$G_0(x, y) = Ae^{\frac{-(x-\mu)^2}{2\sigma_x^2} + \frac{-(y-\mu)^2}{2\sigma_y^2}} \quad (1)$$

4.2. Thresholding

Thresholding, which is a simple segmentation method, is then carried out. Thresholding is applied to obtain a binary image from the gray scale image. Thresholding technique compares each pixel intensity value (I) with respect to the threshold value (T). If I<T, the particular pixel is replaced with a black pixel and if I>T, it is replaced with a white pixel. A threshold value (T) of 127 is used in our work which classifies the pixel intensities in the gray scale image. Maximum value of 255 is the pixel value used if any given pixel in the image passes the threshold value. The two types of thresholding that are implemented are Inverted Binary Thresholding and Otsu's Thresholding. Inverted Binary Thresholding inverts the colors, to be white image in a black background. This thresholding operation can be expressed as shown in Eqn. 2.

$$Dest(x, y) = \begin{cases} 0 \text{ if } src(x,y) > T \\ \max Val(255) \text{ otherwise} \end{cases} \quad (2)$$

So, if the pixel intensity src(x, y) is greater than the threshold value T, then the new intensity of the pixel is initialized to 0. Otherwise, the pixels are set to max*Val*.

Nobuyuki Otsu has given us the Otsu's method[20]. Clustering-based image thresholding is achieved from this method. Otsu binarization automatically calculates a threshold value from image histogram for a bimodal image, which is an image whose histogram has two peaks. In Otsu's method we try to find the threshold that minimizes the intra-class variance (the variance within the class), defined as a weighted sum of variances of the two classes as seen in Eqn. 3. Weights ω_0 and ω_1 are the probabilities of the two classes separated by a threshold t and σ_0^2 and σ_1^2 are variances. The class probability $\omega_{0,1}(t)$ is computed from the L histograms. This is shown in Eqn. 4.

$$\sigma_\omega^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{3}$$

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \tag{4}$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i)$$

Otsu shows that minimizing the intra-class variance and maximizing inter-class variance generates the same results as seen below in Eqn. 5.

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_0(\mu_0 - \mu_T)^2 + \omega(\mu_1 - \mu)^2 \tag{5}$$
$$= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2$$

This is expressed in terms of for probabilities and for means. While the class mean $\mu_{0,1,T}(t)$ can be expressed as shown in Eqn.6.

$$\mu_1(t) = \sum_{i=t}^{L-1} i \frac{p(i)}{\omega_1} \tag{6}$$
$$\mu_T = \sum_{i=0}^{L-1} ip(i)$$

The following relations in Eqn. 7 can be easily verified.

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T \tag{7}$$
$$\omega_0 + \omega_1 = 1$$

The class probabilities and means can be computed iteratively. This can provide an effective algorithm.

Before finding contours, threshold has been applied to the binary image to achieve higher accuracy. The below image Fig. 4 shows the front end window that portrays the thresholded version of the user's gesture input.

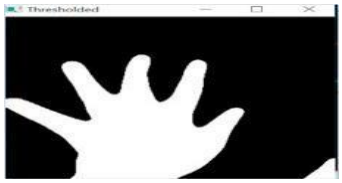

Fig. 4 Front end window that shows the thresholded version of the input gesture

4.3. Contour Extraction

Contours are a useful tool for object detection and recognition in image processing. In our work, we have used contours, to detect and recognize the hand from the background. The curves that link continuous points, which are of the same color, are called contours. Finding the contours is the first step which is like finding white object from

black background in OpenCV. Hence, Inverted Binary Thresholding has been utilized during thresholding. The second step is to draw the contours which can be used to draw any shape provided the boundary points are known. Some gestures in our recognition system with their appropriate contours are shown in the below Fig. 5.

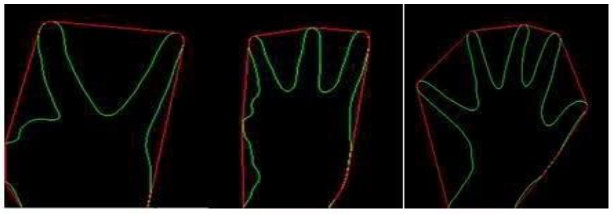

Fig. 5 Contour extraction

4.4. Convex hull and Convexity defects

Mathematically, convex hull of a set *X* of points in any affine space is defined as the smallest convex set that contains *X*. Any deviation of the object from this convex hull can be considered as convexity defect. The convex hull of a finite point set S can be defined as the set of all convex combinations of its points. In a convex combination, each point x_i in S is assigned a weight α_i and these weights are used to compute an average of the points. For each choice of weights, the resulting convex combination is a point in the convex hull. Convex hull can be represented mathematically as shown in Eqn. 8.

$$Convex(S) = \left\{ \sum_{i=1}^{|S|} \alpha_i x_i \mid (\forall i : \alpha_i \geq 0) \wedge \sum_{i=1}^{|S|} \alpha_i = 1 \right\} \quad (8)$$

4.5. Haar Cascade Classifier

For gestures like palm and fist where there are no convexity defects, Haar cascade classifier is used. A collection of positive images, a minimum of 10 original images, taken at different lighting conditions and angles is used. Each of the original images are cropped to contain only the object of interest. Collection of negative images, which doesn't contain the object of interest, a minimum of 1000 images is required. A description file for negative images is created by using create samples library. Each positive image is superimposed on a minimum of 200 images. A vector file is created based on superimposed images (vector file should contain a minimum of 1500 images). Haar training will utilize a minimum of 100 images of size 20 * 20 and the training also can consist of 15 or more stages. The generated XML file is used as cascade classifier to detect objects in OpenCV.

5. Implementation and results

In our gesture recognition system we have included a total of seven gestures, where six of them are static gestures and one is a dynamic gesture. These static gestures are shown in the figure below Fig. 6. The captions written at the top of each gesture i.e. “1”, “2” denotes the number of convexity defects in each gesture. In gestures that do not have any defects i.e. fist, palm, their name has been written as a caption above the gesture.

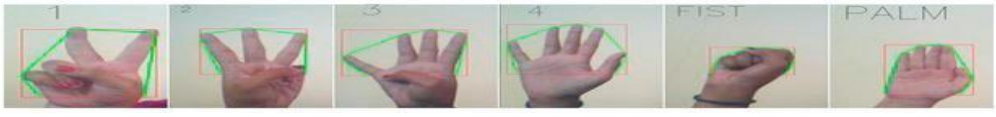

Fig. 6 The static gestures used in the gesture recognition system

The first gesture from the left is a “V” sign or a number two sign which launches the VLC Media Player application as shown in Fig. 7. (a). The second is a number three gesture and it launches Google home page within the user’s default browser as shown in Fig. 7. (b) and the third gesture which is a number four gesture launches YouTube home page. The fourth gesture is a number five gesture or an open palm gesture which in our system closes the application that is running in the foreground. The fifth gesture in the above image is a closed fist that

launches Microsoft PowerPoint. The sixth and final static gesture is a closed palm which toggles the Wi-Fi of the computing apparatus.

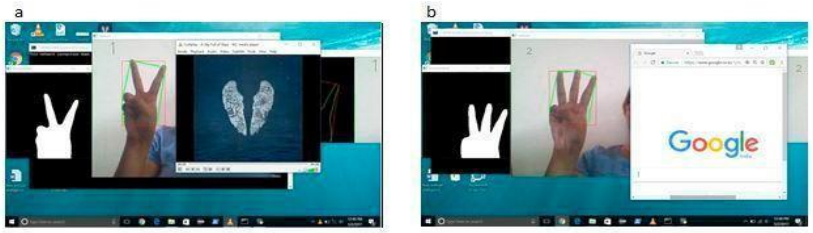

Fig. 7 (a) Gesture “V” launching VLC media player (b) Gesture “3” launching Google home page in browser

In addition to the above mentioned static gestures, the model also has provision for a dynamic gesture. When a moving closed palm gesture is recognized for 5 continuous frames, it is considered to be a dynamic swipe motion. It is used when Microsoft PowerPoint is running in the foreground, to swipe to the next slide within the presentation.

Our first approach to create a gesture recognition system was through the method of background subtraction. Background subtraction, as the name suggests, is the process of separating foreground objects from the background in a sequence of video frames. It is a widely used approach for detecting moving objects from static cameras. When implementing the recognition system using background subtraction, we encountered several drawbacks and accuracy issues. Background subtraction cannot deal with sudden, drastic lighting changes leading to several inconsistencies. This method also requires relatively many parameters, which needs to be selected intelligently. Due to these complications faced, we made a decision to utilize contours, convexity defects and Haar cascade to detect the object (hand). The combination of these methods enabled us to achieve a greater range of accuracy and overcome the challenges faced during the use of background subtraction. To compute the accuracy of our system, we conducted two sets of evaluations. In the first set of evaluation, we used environments which contained different kinds of plain backgrounds without any inconsistencies. In the second evaluation, we used backgrounds with several inconsistencies. Each gesture was performed 10 times in both the environmental setups. The average of the number of times a particular gesture was recognized correctly was taken as its accuracy in percentage and the accuracy obtained is shown in table 1.When implemented against any plain background, the gesture recognition system was robust and performed with good accuracy. This accuracy was maintained irrespective of the colour of the background, provided it is a plain, solid colour background devoid of any inconsistencies. In cases where the background was not plain, the objects in the background proved to be inconsistencies to the image capture process, resulting in faulty outputs. Thus, the accuracy was not as good, in scenarios with plain background. After observing the results produced by the gesture recognition system in different backgrounds, it is recommended that this system be used with a plain background to produce the best possible results and great accuracy.

Table 1 Accuracy of each gesture with plain background and non -plain background

Gesture	Accuracy with plain background(in %)	Accuracy with non-plain background(in %)
“2 finger gesture” (1 convexity defect)	94	40
“3 finger gesture” (2 convexity defects)	93	50
“4 finger gesture” (3 convexity defects)	92	48
“5 finger gesture” (4 convexity defects)	92	52
Palm	95	92
Fist	95	92
Swipe (dynamic)	85	80

\

6. Conclusion and future work

We were able to create a robust gesture recognition system that did not utilize any markers, hence making it more user friendly and low cost. In this gesture recognition system, we have aimed to provide gestures, covering almost all aspects of HCI such as system functionalities, launching of applications and opening some popular websites. In future we would like to improve the accuracy further and add more gestures to implement more functions. Finally, we target to extend our domain scenarios and apply our tracking mechanism into a variety of hardware including digital TV and mobile devices. We also aim to extend this mechanism to a range of users including disabled users.

References

[1] Granit Luzhnica, Elizabeth Lex, Viktoria Pammer. A Sliding Window Approach to Natural Hand Gesture Recognition using a Custom Data Glove. In: 3D User Interfaces (3DUI); 2016 IEEE Symposium on ; 2016 Mar 19 ; New York : IEEE; 2016 ; p.81-90.

[2] Ji-Hwan Kim,Nguyen Duc Thang,Tae-Seong Kim. 3-D hand Motion Tracking and Gesture Recognition Using a Data Glove. In; Industrial Electronics; 2009 IEEE International Symposium on ; 2009 July 5; New York : IEEE;2009 ; p.1013-1018.

[3] Hung CH, Bai YW, Wu HY. Home outlet and LED array lamp controlled by a smartphone with a hand gesture recognition. In: Consumer Electronics (ICCE); 2016 IEEE International Conference on ; 2016 Jan 7; New York : IEEE;2016 ; p.5-6.

[4] Hung CH, Bai YW, Wu HY. Home appliance control by a hand gesture recognition belt in LED array lamp case. In: Consumer Electronics (GCCE); 2015 IEEE 4th Global Conference on ; 2015 Oct 27; New York : IEEE;2015; p. 599-600

[5] She Y, Wang Q, Jia Y, Gu T, He Q, Yang B. A real-time hand gesture recognition approach based on motion features of feature points. In: Computational Science and Engineering (CSE); 2014 IEEE 17th International Conference on; 2014 Dec 19; New York: IEEE;2014;p.1096-1102.

[6] Lee DH, Hong KS. A Hand gesture recognition system based on difference image entropy. In: Advanced Information Management and Service (IMS), 2010 6th International Conference on; 2010 Nov 30; Seoul; New York: IEEE; 2010 ; p. 410-413.

[7] Dulayatrakul J, Prasertsakul P, Kondo T, Nilkhamhang I. Robust implementation of hand gesture recognition for remote human-machine interaction. In :Information Technology and Electrical Engineering (ICITEE); 2015 7th International Conference on ;2015 Oct 29; p.247-252.

[8] Tsai TH, Huang CC, Zhang KL. Embedded virtual mouse system by using hand gesture recognition. In: Consumer Electronics-Taiwan (ICCE-TW); 2015 IEEE International Conference on; 2015 Jun 6; Taiwan, Taipei; New York: IEEE; 2015; p. 352-353.

[9] Hussain I, Talukdar AK, Sarma KK. Hand gesture recognition system with real-time palm tracking. In: India Conference (INDICON);2014 Annual IEEE ;2014 Dec 11; India, Pune; New York: IEEE; 2014; p. 1-6.

[10] Huong TN, Huu TV, Le Xuan T. Static hand gesture recognition for vietnamese sign language (VSL) using principle components analysis. In: Communications, Management and Telecommunications (ComManTel); 2015 International Conference on; 2015 Dec 28; p. 138-141.

[11] Chen Y, Luo B, Chen YL, Liang G, Wu X. A real-time dynamic hand gesture recognition system using kinect sensor. In: Robotics and Biomimetics (ROBIO); 2015 IEEE International Conference on ; 2015 Dec 6; New York : IEEE;2015; p. 2026-2030.

[12] C. Wang, Z. Liu and S. C. Chan. Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera. *IEEE Transactions on Multimedia* 2015; **17**(1): 29-39.

[13] Chen WL, Wu CH, Lin CH. Depth-based hand gesture recognition using hand movements and defects. In: Next-Generation Electronics (ISNE); 2015 International Symposium on; 2015 May 4 ; Taiwan, Taipei ; New York : IEEE;2015 ;p. 1-4.

[14] Wong WS, Hsu SC, Huang CL. Virtual touchpad: Hand gesture recognition for smartphone with depth camera. In: Consumer Electronics-Taiwan (ICCE-TW);2015 IEEE International Conference on; 2015 Jun 6; Taiwan, Taipei; New York : IEEE;2015 ; p. 214-215.

[15] Ishiyama H, Kurabayashi S. Monochrome glove: A robust real-time hand gesture recognition method by using a fabric glove with design of structured markers. In: Virtual Reality (VR); 2016 IEEE ;2016 Mar 19; Greenville, SC; New York : IEEE;2016;p. 187-188.

[16] Suriya R, Vijayachamundeeswari V. A survey on hand gesture recognition for simple mouse control. In: Information Communication and Embedded Systems (ICICES); 2014 International Conference on; 2014 Feb 27; India, Chennai; New York : IEEE;2014;p. 1-5.

[17] Chanda K, Ahmed W, Mitra S. A new hand gesture recognition scheme for similarity measurement in a vision based barehanded approach. In:Image Information Processing (ICIIP); 2015 Third International Conference on; 2015 Dec 21; ; New York : IEEE;2015 ; pp. 17-22.

[18] Luzhnica G, Simon J, Lex E, Pammer V. A sliding window approach to natural hand gesture recognition using a custom data glove. In :3D User Interfaces (3DUI); 2016 IEEE Symposium on; 2016 Mar 19; Greenville, SC ; New York : IEEE;2016 ;p. 81-90.

[19] Chen Y, Ding Z, Chen YL, Wu X. Rapid recognition of dynamic hand gestures using leap motion. In: Information and Automation; 2015 IEEE International Conference on; 2015 Aug 8; New York : IEEE;2015 ;p. 1419-1424.

[20] Otsu, Nobuyuki. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* 1979; **9**(1): 62-66.