Hand-Gesture Recognition-Algorithm based on Finger Counting

M. Perimal, S.N. Basah, M.J.A. Safar and H. Yazid
School of Mechatronic Engineering, Universiti Malaysia Perlis, Arau Perlis, Malaysia.
andymuni@rocketmail.com

***Abstract*—The concept of hand gesture recognition has been widely used in communication, artificial intelligence, and robotics. The most contributing reason for the emerging gesture recognition is that they can create a simple communication path between human and computer called HCI (Human-Computer Interaction). Therefore, a hand gesture recognition algorithm was developed for fourteen hand gestures based on finger counting. The algorithm counts fingers and recognizes gesture based on the maximum distance between the fingers detected. The algorithm divided into four main parts: image acquisition, pre-processing, finger detection, and gesture recognition. The experimental results show that the algorithm can count fingers accurately and recognize 10 gestures (associated with 1, 2, 3 and 5 fingers) with good performance (70 to 100 percent of successful detection) and 4 gestures (associated with 4 fingers) with average performance (50 to 70 percent of successful detection). Additionally, the algorithm was tested under variation of the scene and dynamic parameters, to understand its performance further.**

***Index Terms*—Finger Detection; Hand Gesture Recognition; Human-Computer Interaction (HCI); Pre-Processing.**

I. INTRODUCTION

The gesture is a verbal or non-verbal communication which conveys a message to others in the form of hand or head movements [1]. Therefore, researchers nowadays apply the same concept to develop an algorithm to establish a human-computer interaction which accepts the hand gesture as an input to give an output [2].

This concept can be used to replace the human interface devices (HID) which used to control a device. A human interface device or (HID) is an interface device that takes input from human and at the same time gives output directly to human as well [2]. These HID devices are most familiar to humans like a mouse, keyboard, joysticks and remote control. Its purpose is to control a device to perform a specific task. For instance, mouse and keyboard are used to operate a computer in which the use presses specific buttons to activate specific function [3]. The use of gesture recognition as an HID device will change the perspective human.

In present days, a lot of research has been taking place to develop a simple and natural interaction between human and computer. Image processing techniques can be used to establish a vision-based interaction. Gesture recognition interfaces which are used as a way of natural communication between man and machine give rise to a variety of applications such as equipment free remote-control systems, sign language interpretation and many more. There are several hand gesture algorithms were developed over the years. However, these algorithms limited performances in their practical environment – handling noisy images, small hand or when the distance between hand and camera is large, low light intensity environment and hand with accessories – watch, bracelet, etc. Thus, we have developed an algorithm for hand gesture recognition based on finger counting with the aim to minimize those limitations.

II. LITERATURE REVIEW

There have been many types of research has been developed to recognize a different type of hand gesture. The recognized hand gestures are used to perform various type of application such as controlling a computer software and mobile robot. There are two ways to recognize hand gestures: using the vision-based system or glove-based system.

The glove-based system uses a sensor which can measure the signals of muscle movement for each gesture [3]. Then, the data are classified using a classification software. This method is very accurate as it measures the muscle signal using appropriate sensors. For data acquisition or processing, gloves based system define as array sensors and electronics. This glove is made of Lycra which consists of sensors, records data based on hand motion. Hand motions are described based on joints and the degrees of freedom (DOFs). Thus, the glove is made up of sensors, one sensor for one DOF to be precise, as human hands have three types of DOF. The distal interphalangeal (DIP) and proximal interphalangeal (PIP) joints of each finger have 1 DOF, 2 DOF for the metacarpophalangeal (MCP) joints and trapeziometacarpal (TMCP) joint of the thumb [3] has 3 DOF. Nevertheless, the cost of sensors is approximately high. Apart from that, wearing a glove will be inconvenient. A vision-based system uses vision device such as a camera (Kinect and Leap Motion) to capture the image of the gesture and process it using image processing techniques. There are many methods can be used in this method. The image processing technique is divided into three sub-stage: segmentation and pre-processing method, feature extraction and finally gesture recognition.

Segmentation and pre-processing method are mainly about 1
determining the shape of the hand (contour) in the image. In this process, the contour of the hand in the image will be determined (therefore the hand can be separated from the background image) [4]. A static hand gesture in the input image must be segmented. For dynamic gesture, the gesture must be tracked and located before segmentation [4]. This will be the difference between static and dynamic hand gesture. Before segmentation, the location of the hand must be located using the skin color, and a bounding box is used to identify the hand in the image or frame [5]. The video (from the webcam) is divided into frames, and each frame segmented one by one so that the position of hand can be

tracked [4] or the hand is tracked based on the skin color, shape or even by using a filter such as Kalman filter [4]. There are two methods to model the hand, namely parametric (Gaussian Model (GM) and Gaussian Mixture Model (GMM)) and non-parametric (histogram based techniques) techniques using skin and non-skin pixels [4].

For a successful hand recognition process, feature
2. extraction plays an important role after the segmentation process [5]. Different methods can be used to extract the feature in an image. The shape of the hand can be determined by Self-Growing and Self-Organized Neural Gas (SGONG) neural algorithm which then gives an output on the hand scope, palm region, and center. When the farthest distance from the center of gravity of the segmented image is calculated to count the number of fingers [6] in the segmented hand image by extract one binary signal (1D). In some cases, the segmented image divided into different size of blocks which represents the image brightness measurement.

After the input image, has been model and analyses, gesture classification method can be applied to recognize the gesture.
3. The factor affects recognition process is a selection of features parameter and appropriate classification algorithm [7]. Misclassification may be a result of generating many hand postures, so that edge detection or contour operator are not suitable to be used in gesture recognition [8]. One of a method to classify gesture recognition will be Euclidean distance metric which uses the distance from the center of the hand to the fingers [9]. Other than that, the dynamic gesture can be classified using HMM tool [10] more efficiently as they recognize the gesture and their parameters are studied from the training data [11] compared to Finite State Machine (FSM), Learning Vector Quantization, and Principal Component Analysis (PCA). One of the major tools contribute in this classification fields will be Neural Network for extract hand shape [12] and for hand gesture recognition. Soft computing tools such as Fuzzy C-Means clustering (FCM) [12], Genetic Algorithm Gas can be used for gesture classification. Other than that, Support Vector Machine (SVM) can be used for hand gesture classification as misclassification of gestures is reduced [13].

The vision-based system had some limitation which affects the hand gesture recognition. The limitation is the viability of the technology, limitation on equipment, image noise, hand distance from the camera, camera's resolution and quality. In other words, the hand gesture recognition will affect by image noise, light intensity and hand distance from the camera. These limitations will hinder the algorithm to detect the hand in the image. Most of the researchers used five hand gesture in their research [3]. This is due to the fact that using less hand gesture will increase the accuracy rate of the hand gesture algorithm.

III. ALGORITHM DEVELOPMENT

The proposed algorithm consists of four main parts – image acquisition, pre-processing, finger detection, gesture recognition. Figure 1 shows the overall steps involved in a successful hand gesture recognition algorithm.

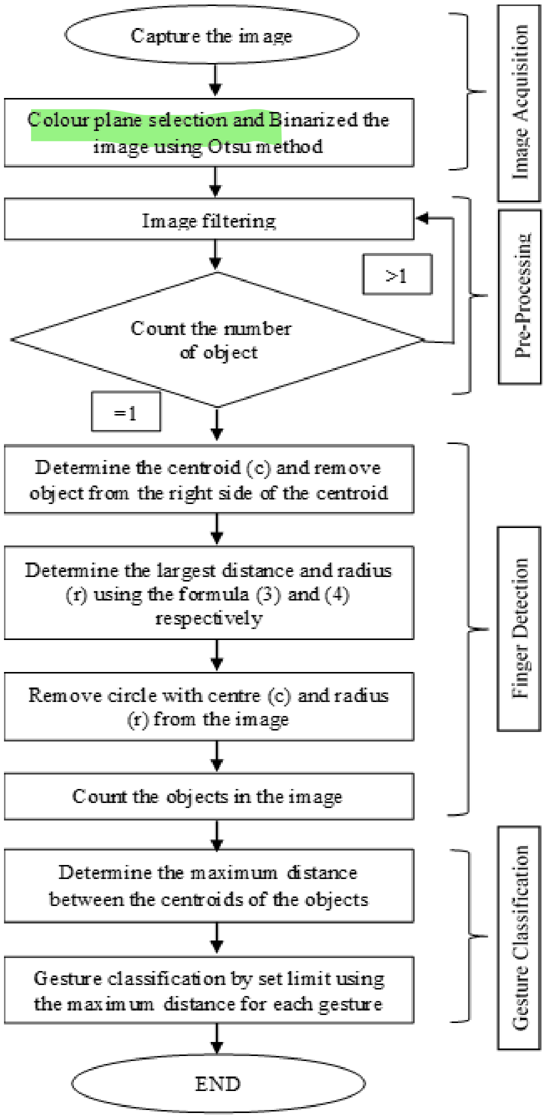

Figure 1: Flowchart for hand gesture recognition algorithm based on finger counting

A. Image Acquisition

Image acquisition is the first step in this system. Images are captured by an off-the-shelf HD webcam with a resolution of 16Mp. This webcam is interfaced with MATLAB 2012 to capture the image. The subject is requested to place their hand on a dark surfaced platform to capture the image. Fourteen hand gesture is used in this research which are obtained from one hand, and the subjects are asked to show the gesture in front of the camera to capture the image. Then, the image is load in the MATLAB 2012 software, and the image is resized half of its original size for the convenient of the user.

After that, the RGB image is converted into YCbCr color space. YCbCr color space is a YUV color space family, and it is also a linear luminance or chromaticity color space. YCbCr stands for Y is luminance, Cb and Cr are chromaticities of blue and red colors. The Cb and Cr are two-dimensional independents. The formula to convert an RGB color space to YCbCr color space is:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1440 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

Then, the image is converted into binary image using Otsu's method. Otsu's method is used to automatically perform clustering-based image thresholding or the reduction of a gray level image to a binary image. In other words, it is simply an algorithm for segmentation of object regions. A simple but effective tool to separate objects from the background. Otsu method principle one simply defines the within-class variance. The threshold value will be determined by the algorithm using Equation (2). Therefore, the user does not have to determine the threshold value manually.

$$\sigma^2_{Within}(T) = \omega_B(T)\sigma^2_B + \omega_0(T)\sigma^2_0(T) \tag{2}$$

where: $\omega_B(T)$ = $\sum_{i=0}^{T-1} p(i)$, [0, L-1] is the range of intensity

$\omega_0(T)$ = $\sum_{i=0}^{L-1} p(i)$

$\sigma^2_B(T)$ = Variance of the pixels in the background (below threshold)

$\sigma^2_0(T)$ = Variance of the pixels in the foreground (above threshold)

B. Pre-Processing

After a successful binary image is determined, the image should be cleaned to process further. First, the small objects in the image must be removed. Thus, objects which are less than a certain pixel must be removed to obtain the object that the user needed. If the object has any holes, it should be fixed. This will ensure a good hand gesture recognition. Finally, the object inside the image is counted. If the number of the object is one (make sure that only the hand is detected in the image), then the algorithm can proceed further.

C. Finger Detection

First, the centroid of the binary image is determined. This will be the center of the hand which will be used to remove the wrist of the hand and palm. After finding the centroid of the image, the right side of the image which is the wrist will be removed as it does not involve the finger detection. Then, the largest distance between one pixel to another on the contour of the object is determined using the equidistance formula. The formula of equidistance is:

$$dist = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{3}$$

$$radius = dist \div 2 \tag{4}$$

where: (x_1, y_1) = Coordinate of a pixel

(x_2, y_2) = Coordinate of a pixel

The maximum distance will be divided by two to obtain the radius of the circle. A circle will be drawn in the image, and it will be removed (palm) as we obtain the radius and center of the circle. Moreover, the radius should times by 1.26 to make the circle bigger so that only the fingers can be obtained. The number of an object inside the image will be calculated and displayed.

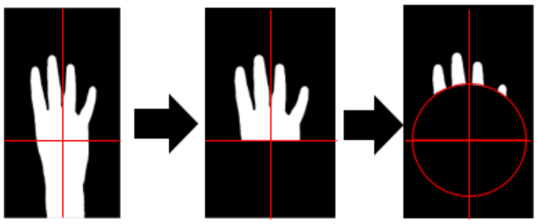

Figure 2: Finger detection for four finger counts

D. Gesture Classification

The detected fingers are classified into their respective finger count. The gestures are classified using the maximum distance between the centroid of the two fingers determined in the finger detection process. The distance between each centroid of the two fingers is determined using the equidistance Equation (3).

IV. TESTING AND ANALYSIS

This experiment is conducted with five subjects. For this experiment, a high definition camera was used to capture the image of hand gesture. This is because a low definition camera will produce more noise in the image which difficult to process. Fourteen hand gestures according to Figure 3 were used in this experiment, and 10 data were collected for each subject. Thus, a total of 700 data was collected in this experiment (14x5x10).

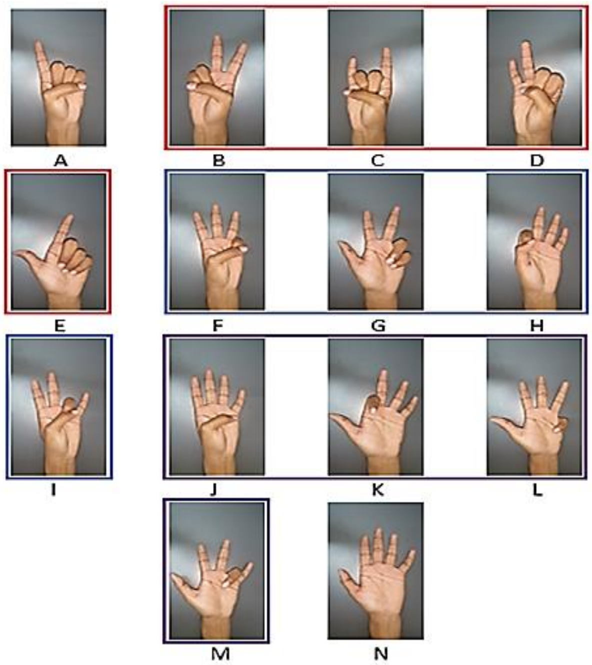

Figure 3: Hand gesture used in the experiment. (A and N uses one gesture to represent 1 and 5 finger counts, B to M uses four gestures to represent 2, 3 and 4 finger count)

In this experiment, one gesture was used for finger count one and five. Whereas, four gestures are used for finger count two, three and four. The images were taken in a controlled lighting environment (room light) to obtain a good result. The distance between the hand and the camera is between 15 to 20 cm. The data are collected for single hand recognition. Images are captured by an off-the-shelf HD webcam with a resolution of 16Mp to avoid noise in the image. Figure 4 shows the process flow of hand gesture recognition based on finger counting.

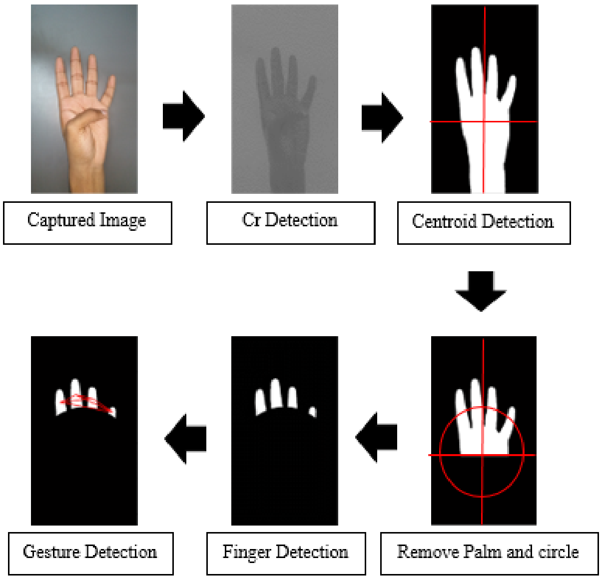

Figure 4: Graphical flow chart for hand gesture recognition algorithm based on finger counting

The algorithm was tested using different parameters to verify the performance of the hand gesture recognition algorithm. The parameters that chosen in this experiment were light intensity, size, noise, and effect of selective hand accessories.

V. RESULT AND DISCUSSION

The experiment was carried out for one to five finger counts as shown in Figure 2. Five subjects were chosen for this experiment. The result was collected based on the number of successful hand gesture recognition for every finger count. Figure 5 shows the percentage of successful finger detection and gesture recognition for the algorithm that has developed.

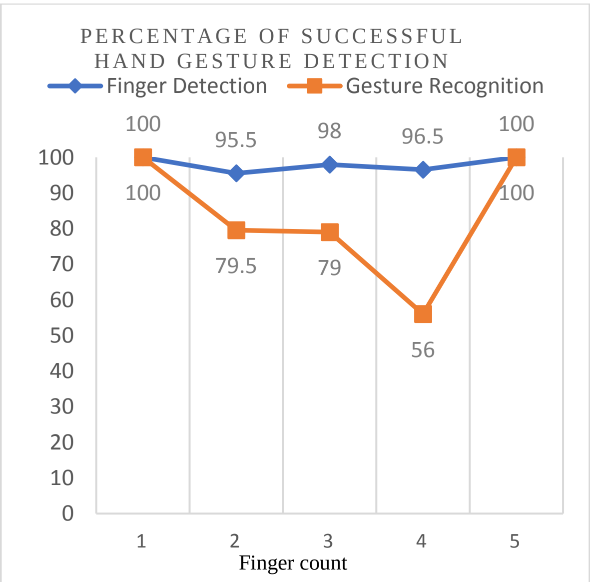

Figure 5: Percentage of successful hand gesture detection

The results show that the algorithm can correctly identify the number of fingers. This is indicated by high values of detection accuracies – more than 90% – in all cases for 1 to 5 fingers. The percentage of successful finger detection for finger count two is lower than the other finger counts because gesture 4 (refer Figure 3 image E) fail to detect correctly. When the circle is removed from the filtered image of finger count two for gesture 4, the thumb cannot be detected.

Additionally, the algorithm can produce 100% correct gesture recognitions for 1 and 5 finger count and good recognition (around 80% correct) for 2 and 3 finger counts. We also observed, the average (around 56%) gesture recognition accuracy for 4 finger counts.

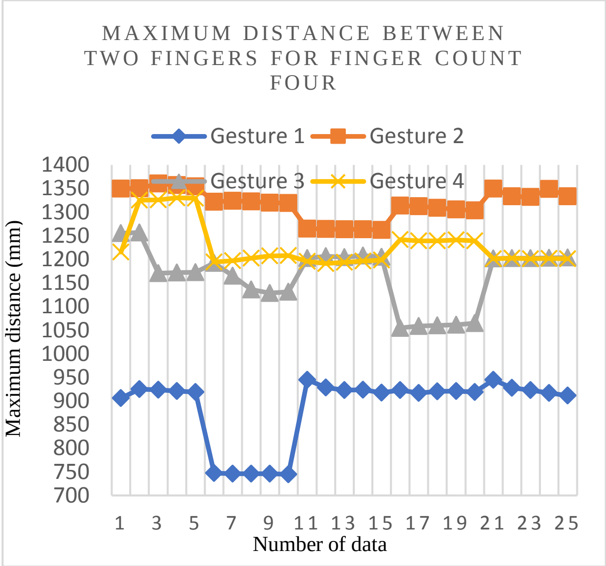

Figure 6: Maximum distance between fingers for four finger counts

Figure 6 shows that, 5 data of the maximum distance between fingers for four finger count which is collected from 5 subjects. It can be observed that gesture one and two can be easily classified as the value is far apart from one another. Whereas, there is an overlap for gesture three and four. This is mainly because the failure in finger detection which makes the maximum distance similar to other gestures.

The performance of the hand gesture recognition algorithm was tested by varying some parameters in the algorithm. The parameters are noise, light intensity, size of the hand and effect of selective hand accessories. Figures 7 and 8 show the relationship between the successful hand gesture detection and the varied parameters.

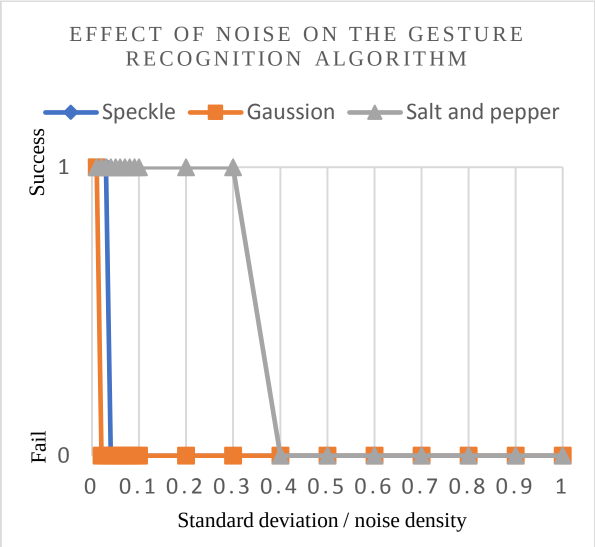

Figure 7: Effect of noise on the gesture recognition algorithm

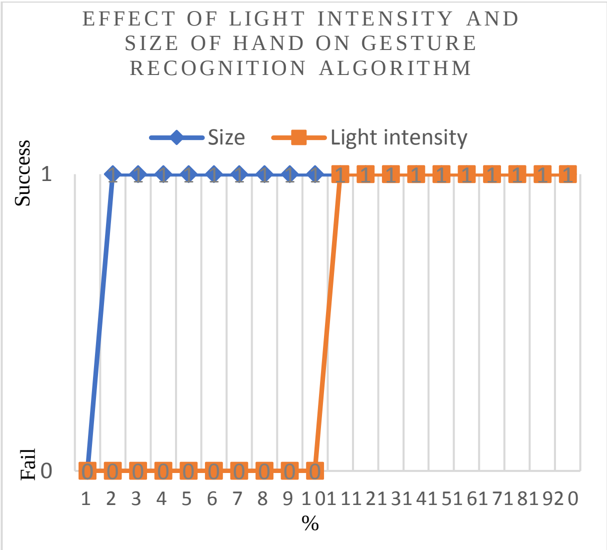

Figure 8: Effect of light intensity and size of a hand on the gesture recognition algorithm

A. *Noise*

Three types of noise were tested in this experiment, which are Gaussian noise, salt and pepper noise and speckle. We observed that, when the magnitude of noise in the image increases, it became more difficult to produce correct gesture recognition. For example, when salt and pepper noise with of more than 0.4 noise density were added as shown in Figure 9, the accuracy of the gesture recognition algorithm decreases by 50% due to the changes in the maximum distance between fingers. Additionally, for Gaussian noise, a mean of 1 and standard deviation more than 0.02 will reduce the accuracy of the algorithm by 90%. Finally, the accuracy of gesture recognition reduces by 90% when the standard noise deviation of more than 0.03 for speckle noise.

Figure 9: Effect of salt and pepper noise at 0.4 noise density on hand gesture recognition algorithm

B. *Light Intensity*

The brightness of the image was changed to study the effect of light intensity on hand gesture recognition algorithm. If the brightness of the image is increased, the hand gesture algorithm works perfectly. Whereas a decrease in the brightness of the image below 0.1% will decrease the percentage of successful hand gesture detection by 50%. The image cannot be detected if the brightness is below 0.01%.

C. *Hand Size*

The size of the image is changed using the MATLAB. The original image size is decreased to study the effect of hand size on hand gesture recognition algorithm. The gesture recognition reduces to 90% after the image is resized at a value of 0.01 from the original size of the image as shown in Figure 10.

Figure 10: Effect of size on the hand gesture recognition algorithm

D. *Hand Accessories*

We also have analysed the effect of common hand accessories such as watch and bracelet on gesture recognition performance. We observed that watch give a huge effect on the hand gesture recognition algorithm. The YCbCr color space detection fails to detect the silver watch. Thus, the watch separates the wrist and palm of the hand as shown in Figure 11. This is due to the limitation of using YCbCr color plane.

(a) (b)

Figure 11: Effect of the watch on the hand gesture recognition algorithm (a) the original image and (b) the binary image

However, the prayer threads which commonly worn by Indians does not affect the gesture recognition algorithm. The YCbCr color plane successfully detects the red color thread as shown in Figure 12.

(a) (b)

Figure 12: Effect of prayer thread on the hand gesture recognition algorithm (a) the original image and (b) the binary image

VI. Conclusion

A hand gesture recognition algorithm was developed for fourteen hand gestures based on finger counting. The algorithm counts fingers and recognized gestures using the maximum distance between the centroid of the fingers. The experimental results in Figure 5 shows that the algorithm can count finger accurately and recognize 10 gestures (associated with 1, 2, 3 and 5 fingers) with good performance and 4 gestures (associated with 4 fingers) with average performance. Additionally, the algorithm as tested under variation of the scene and dynamic parameters, to further understand its performance.

References

[1] Takuichi Nishimura, Hiroaki Yabe, Ryuichi Oka, Toshiro Mukai, "Recognition of Gestures Using Morphological Features of Networks Made of Gesture Motion Images and Word Sequences", vol. 00, no., pp. 39, 1999.

[2] M. A. Rashid and X. Han, “Gesture Control of ZigBee Connected Smart Home Internet of Things,” pp. 667–670, 2016.

[3] N. H. Adnan, K. Wan, and A. B. Shahriman, “Measurement of the Flexible Bending Force of the Index and Middle Fingers for Virtual Interaction,” vol. 41, no. Iris, pp. 388–394, 2012.

[4] G. Qh, N. Cpf, Q. Pikpggtkpi, G. Qh, N. Cpf, Q. Pikpggtkpi, O. Ucokv, C. T. K. Iockn, O. Fkrcmmwoct, and I. Iockn, “Static Hand Gesture Recognition Using Mixture of Features and SVM Classifier,” pp. 1094–1099, 2015.

[5] Z. Y. Meng, J. S. Pan, K. K. Tseng, and W. Zheng, “Dominant points based hand finger counting for recognition under skin color extraction in hand gesture control system,” Proc. - 2012 6th Int. Conf. Genet. Evol. Comput. ICGEC 2012, pp. 364–367, 2012.

[6] M. M. Hasan and P. K. Mishra, “Robust Gesture Recognition Using Gaussian Distribution for Features Fitting,” vol. 2, no. 3, 2012.

[7] E. Stergiopoulou and N. Ã. Papamarkos, “Engineering Applications of Artificial Intelligence Hand gesture recognition using a neural network shape fitting technique,” Eng. Appl. Artif. Intell., vol. 22, no. 8, pp. 1141–1158, 2009.

[8] J. J. L. Jr., A Survey of Hand Posture and Gesture Recognition Techniques and Technology, vol. 102.

[9] H. Zhou, D. J. Lin, and T. S. Huang, “Static Hand Gesture Recognition based on Local Orientation Histogram Feature Distribution Model,”

[10] A. A. Hassan and S. N. Basah, “Gesture-based remote-control system using coordinate features,” vol. 11, no. 8, pp. 4979–4986, 2016.

[11] M. M. Gharasuie and H. Seyedarabi, “Real-time Dynamic Hand Gesture Recognition using Hidden Markov Models,” pp. 194–199, 2013.

[12] M. Murugeswari and S. Veluchamy, "Hand gesture recognition system for real-time application," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, 2014, pp. 1220-1225.

[13] X. Sun, L. Liu, H. Wang, W. Song, and J. Lu, “Image Classification via Support Vector Machine,” no. Iccsnt, pp. 485–489, 2015.