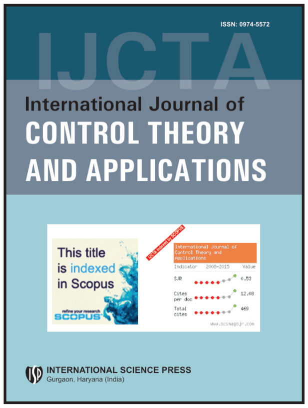

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 9 • Number 43 • 2016

Static Hand Gesture Recognition Using Accurate End Point Identification (AEPI) Method

A.V. Dehankar[a], Sanjeev Jain[b] and V.M. Thakare[c]

[a]Assistant Professor, Department of Computer Technology, Priyadarshini College of Engineering, Nagpur. Email: archana_dehankar@rediffmail.com
[b]Vice-Chancellor, Shri Mata Vaishno Devi University, Jammu. Email: dr_sanjeevjain@yahoo.com
[c]Professor and Head of Deptt., CSE, Amravati University, Amravati. Email: vilthakare@yahoo.co.in

Abstract: In today's world, communication of humans with computers and other gadgets is increasing rapidly. The methods and modes of communication are changing day by day. New ways of interacting with computers are designed. The Hand Gesture Recognition system is useful to detect the gestures with the help of image capturing devices installed on computer and other gadgets. The rapid growth and improvement in mobile phones in past few years have led to innovations in visualization and interaction with devices. Even though the touchscreens have enhanced interactivity with user but in future user may request for more natural inputs like Hand Gesture. Currently there are technologies using Hand Gesture Recognition Systems, but the method we have implemented is more accurate than other present systems. This paper describes the working of Accurate End Point Identification (AEPI) method which is used for Hand Gesture Recognition. The method is based on Morphological Computations. The whole process to identify the Hand Gesture is done on the machines. This approach can be easily used for the static systems and also for real time systems.

Keywords: Hand Gesture Recognition, Accurate End Point Identification (AEPI).

1. INTRODUCTION

Human gestures add value to voice communication during person to person communication. In fact sometimes hand gestures are able to communicate the complete message without voice. Thus hand gestures carries a potential to provide a new dimension for establishing communication among the persons who are not able to speak native languages of each other. Most of the computing devices have interfaces for human machine interaction that allows user to interact with these computing devices using keyboard, mouse, joystick, touch pad etc. These interfaces are limited to few functionalities and interactions like navigation, menu control, software/hardware control etc. Also, each of these interaction devices and methods have their own pros and cons. Researchers are trying to utilize hand gestures as a medium of interaction with computing devices. The hand gestures can be used to control variety of devices locally and remotely. Previously, Computer Vision software was limited only to desktop computers. Enhancement in mobile hardware devices makes it possible to utilize Hand Gesture Recognition on mobile platform by using image processing and computer vision software.

Hand gesture recognition systems can be applied in controlling software and hardware in desktop environment. In robotics ,various tasks performed by robots can be controlled through hand gestures. Controlling mobile applications, playing games etc. are also possible. This study focuses on learning hand gesture recognition systems. Gesture recognition systems are implemented in certain typical steps including gesture acquisition, segmentation, feature extraction and recognition of gesture. Each stage involves certain work to be carried out to correctly identify the gesture. In this paper ,we proposed a method for Hand Gesture Recognition Using Accurate End Point Identification (AEPI).

2. RELATED WORK

In this section, a review of most recent work carried out in the field of hand gesture recognition is discussed. Segmentation is an important process in hand gesture recognition in which the objects including hand gesture are segmented into a regions. The type of input decides when to apply the segmentation process. In case of static gesture input image segmentation is directly applied whereas for dynamic gesture input the gesture is located and tracked [1].

Serban Oprisescu and et. al., has developed an automatic algorithm for gesture recognition which works on static images using the depth and intensity information and for classification decision tree is used, the system provided about 93.3% recognition rate [2].

Ankita Chavda and et. al., has presented a system that gives approximately 81% accuracy on ASL recognition. A scale invariant algorithm is implemented using rule based approach for alphabet recognition[3].

Jayshree R.Pansare and et. al.. has worked on system to identify Devanagari Sign Language Number System Using Hand Gestures. The authors used discrete Cosine Transform and Edge Orientation Histogram to extract and compare the features. The system works in a simple background [4].

Trong-Nguyen has proposed a promising system with 98% accuracy using artificial neural network for static hand gesture recognition [5]. Gaurav Manik Bidgar and et. al., has implemented a system which uses hand gestures as control commands. The system employed classification algorithm and skin color segmentation for foreground and background [6].

Miss.Neha.B.Bhoyar and et. al., has proposed a new technique based on shape analysis using back propagation learning algorithm in neural networks [7]. Haitham Badi and et. al., has implemented a hand gesture system to perform operations like open, close, maximize, minimize, cut and paste. The authors explored two methods which addresses translation and scaling problems of a gesture image. Complex moments algorithm solves rotation problem along with scaling and translation. For classification, back-propagation learning algorithm is used in this system [8].

Rajashree Patil and et. al., has proposed a system to recognize ASL alphabets from a static gesture. The system employs DWT based feature extraction and minimum vector classifier [9]. Chethana N S and et. al., has developed a system that works in three stages. In the first stage, input images are captured through a webcam at a certain frame rate in real time. Feature extraction method is the second stage that extracts features on the basis of orientation, rotation and scaling. The third stage applies K-Curvature algorithm and edge detection for recognition [10].

R.P. Mihail and et. al., has developed a system that works on sensor’s depth streams combinations which captures 3D points of hand gesture. To extract these points, the authors used segmentation and PCA to calculate invariance in rotation across the X and Y axis [11]. Various gesture recognition techniques, their applications in Interaction of Humans with Vehicles and summary of present automotive hand gesture recognition research is discussed in [22]. Some relevant work and survey on hand gesture systems can be found in [16, 17], and [23,24,25,26].

3. ACCURATE END POINT IDENTIFICATION (AEPI) METHOD

The Method is implemented in five steps: Image acquisition & Preprocessing, Removing unwanted objects, black holes & noise, Centroid Detection, Thinning Process and Gesture Recognition.

Step 1: Image acquisition and preprocessing: Once the static image is captured ,it is pre-processed to generate a binary image. During this step ,the actual input image is converted into Grayscale image which is then dilated and converted to binary to obtain its binary complement for further processing. Figure 1 shows the actual RGB input Image captured through webcam or laptop camera.

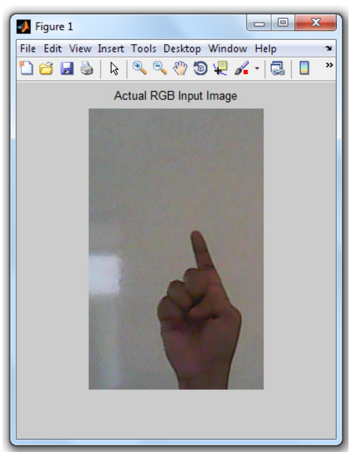

Figure 1: Actual Input RGB Image

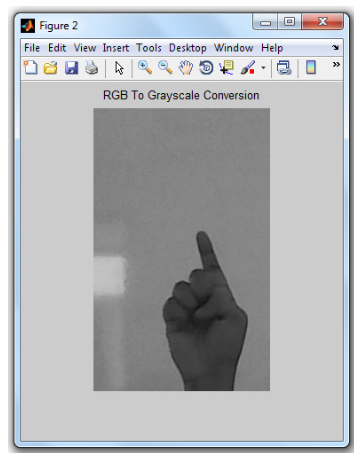

Figure 2: RGB to Grayscale Image Conversion

(a) *RGB to Grayscale Conversion:* The original input image is a RGB image. To extract the correct gesture and identify irrelevant objects present in the image it is essential to convert it into Grayscale intensity image. Grayscale images are helpful in defining the region of interest and to remove the unwanted noise from the image.

As shown in Figure 2, the image is converted into a Grayscale intensity image using the weighted sum of the R, G and B Components as given in equation 1.

$$0.2989 \times R + 0.5870 \times G + 0.1140 \times B \qquad (1)$$

(b) *Dilation of Input RGB Image:* Morphological image processing relies on the ordering of pixels in an image and many times is applied to binary and grayscale images. Through processes such as erosion, dilation, opening and closing, binary images can be modified to the user's specifications [15]. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.

In order to dilate an image, first the structuring element is found in the form of circles for entire image. A nonflat ball-shaped structuring element with radius R = 5 is approximated in XY plane. The center pixel of the structuring element, called the origin, identifies the pixel in the image being processed. and then dilation is applied to smoothen the image.

The dilation is performed using following equation.

$$A \Theta B = \{z \mid (B)_z \subseteq A\}| \qquad (2)$$

The dilated image helps us to locate the hand gesture correctly which is the region of interest for further processing. The dilated image is shown in Figure 3.

C. *Obtaining Binary Image & It's Complement:* Otsu's method [15] is applied for calculating the global threshold value which is used to convert an intensity image to a binary image. Here the threshold value is a normalized intensity value that lies in the range [0, 1].

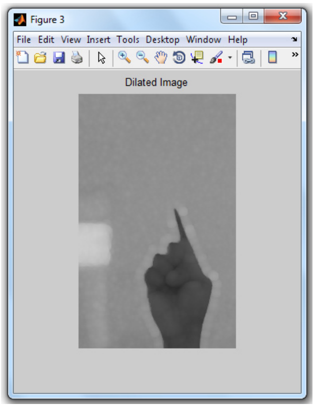

Figure 3: Dlated Image

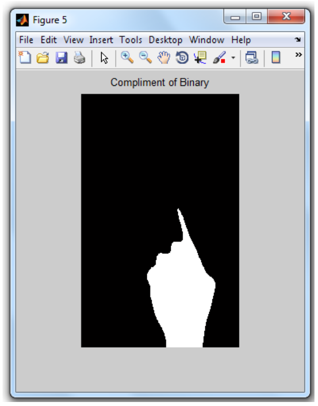

Figure 4: Grayscale to Binary Conversion

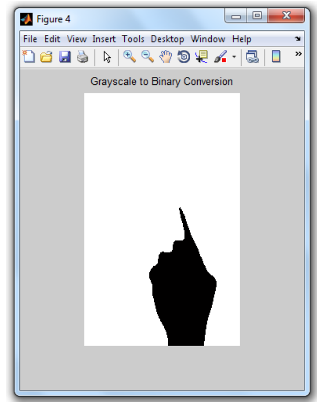

Figure 5: Complement of binary image

This threshold value minimizes the intra-class variance of the black and white pixels and gray scale image is converted to binary image as shown in Figure 4.

The complement of a binary image shown in Figure 4 is computed (figure 5). In the complement of a binary image, zeros become ones and ones become zeros; black and white are reversed. Thus as shown in Figure 5, the hand gesture in the image becomes white and remaining portion becomes black.

Step 2: Removing unwanted objects, black holes & noise: Once the complement of binary image is obtained, it is processed to remove the objects, black holes and noise if present in the image along with hand gesture. Total three object checks are carried out in this step. At every check point, identified objects are colored in RGB scheme to separate them. The first object check(Figure 6) returns the total number of objects present in the image. To identify the objects we have used the procedure given in [18], which returns the number of objects in an image.

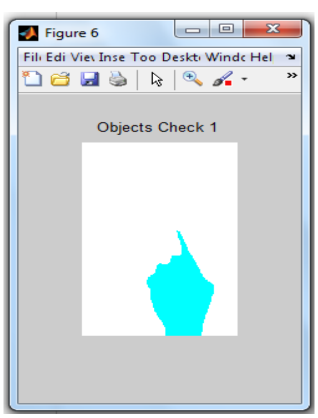

Figure 6: Object Figure Check 1

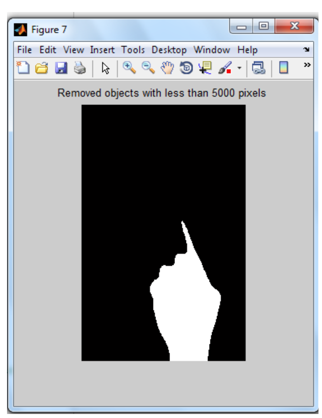

Figure 7: Image after removing objects less than 5000 pixels

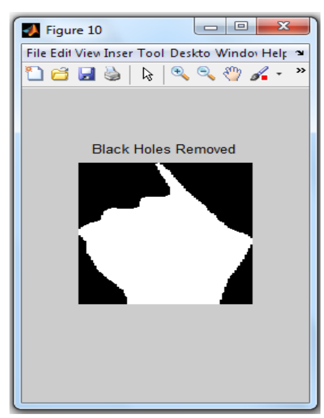

Figure 8: Image after removing black holes

Assuming that hand gesture is the biggest object in image and never has pixels less than 5000, after first object check we remove the objects which are having pixels less than 5000 using the morphological small object removal operation (Figure 7)

There is possibility of having objects whose pixel size is greater than 5000 other than hand gesture. To remove such objects second object check is carried out and if any such objects are present then they are removed. After removing such objects morphological reconstruction function [16] is used to remove the black holes (spot) if any. A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image [16]. The image after removing black holes is shown in Figure 8.

After removing the black holes it is important to remove the noise present in the image if any. To remove the noise and make the image error free Median filtering is used, which is a nonlinear operation and it is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges [17].

Once the noise is removed and image is made error free, third object check is carried out to ensure that only hand gesture remains present in the image for further processing.

Step 3: Centroid Detection: After removing the objects and noise, only hand gesture is retained in the image. The hand gesture includes the palm and fingers. We are interested in counting the fingers and the portion of the palm is not the region of interest. Hence we have to find the point which will help us to divide both the regions. Detecting the centroid of hand gesture will allow dividing it into two regions.

To detect the centroid it is necessary to resize the image into a logical array of 256×256 size. To detect centroid the image region property 'Centroid' – 1-by-Q vector that specifies the center of mass of the region is used. The first element of Centroid is the horizontal coordinate (or x-coordinate) of the center of mass, and the second element is the vertical coordinate (or y-coordinate). Concatenation is performed to return X and Y coordinates of centroid which is in array format. The detected centroid is shown in the Figure 9.

Centroid(1) = X-axis

Centroid(2) = Y-axis

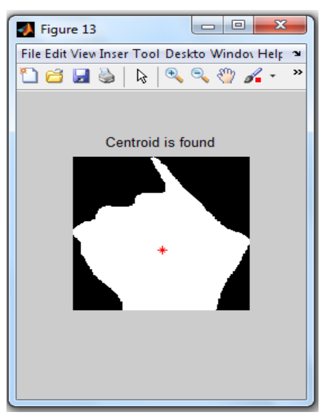

Figure 9: Centroid Detection

Step 4: Thinning Process: After detecting the centroid, thinning operation is performed. Thinning process removes pixels so that an object without holes shrinks to a minimally connected stroke. An object with holes shrinks to a connected ring halfway between each hole and the outer boundary if any. The steps of thinning operation are as given below [18, 19, 20, and 21]:

1. Divide the image into two distinct subfields in a checkerboard pattern.
2. In the first sub iteration, delete pixel p from the first subfield if and only if the conditions G_1, G_2, and G_3 are all satisfied.

3. In the second sub iteration, delete pixel p from the second subfield if and only if the conditions G_1, G_2, and G_3' are all satisfied.

Condition G1:

$$X_H(p) = 1$$

where,

$$X_H(p) = \sum_{i=1}^{4} b_i$$

$$b_i = \begin{cases} 1 \text{ if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0 \text{ otherwise} \end{cases}$$

$x_1, x_2, \ldots, x_8$ are the values of the eight neighbors of p, starting with the east neighbor and numbered in counter-clockwise order.

Condition G2:

$$2 \le \min\{n_1(p), n_2(p)\} \le 3$$

where,

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$$

Condition G3:

$$(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$$

Condition G3':

$$(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$$

The two sub-iterations together make one iteration of the thinning operation. When the user specifies an infinite number of iterations the iterations are repeated until the image stops changing. We get the output as shown in Figure 10.

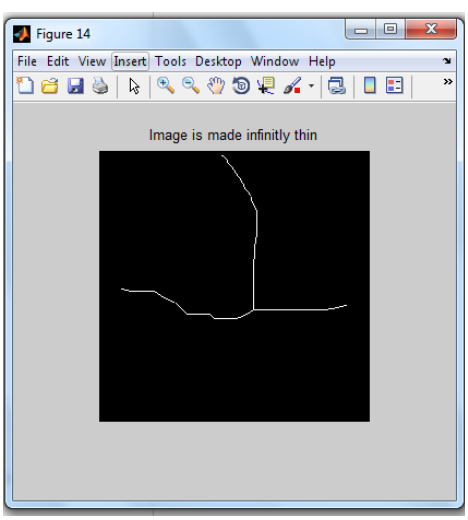

Figure 10: Thinning

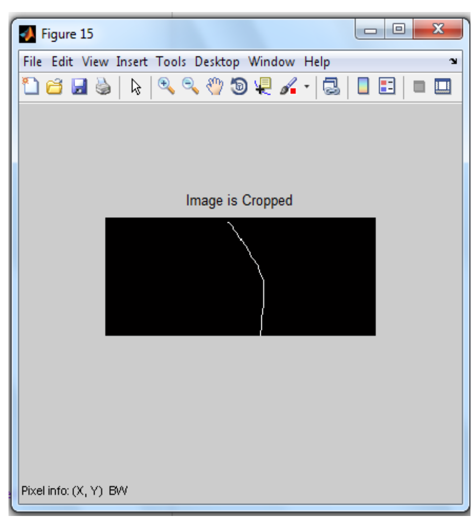

Figure 11: Cropped image

Later the image is cropped on Y-axis from centroid point by 50 pixels so that only finger edges will remain for end point detection. The finger end points are detected to accurately recognize the hand gesture. The cropped image is shown in Figure 11.

Step 5: Gesture Recognition: The end points from cropped image are identified using the morphological operation on a skeleton. Once we get number of end points we can calculate the number of lines (fingers) by dividing the total number of points by two. In this situation, there is possibility of getting more endpoints than actual points of gesture due to additional branches so we have to identify the number of branch points.

To get the number of fingers present in the gesture, following formula is used.

$$\text{Total count of fingers} = ep - bp/2.$$

Where "ep" is number of end points present in the cropped image & "bp" is the number of branch points present in the cropped image.

As shown in Figure 12 there are two end points for the above input gesture image. Sometimes branch points will not be present due to the clarity of gesture in the image. The message is displayed as shown in Figure 13.

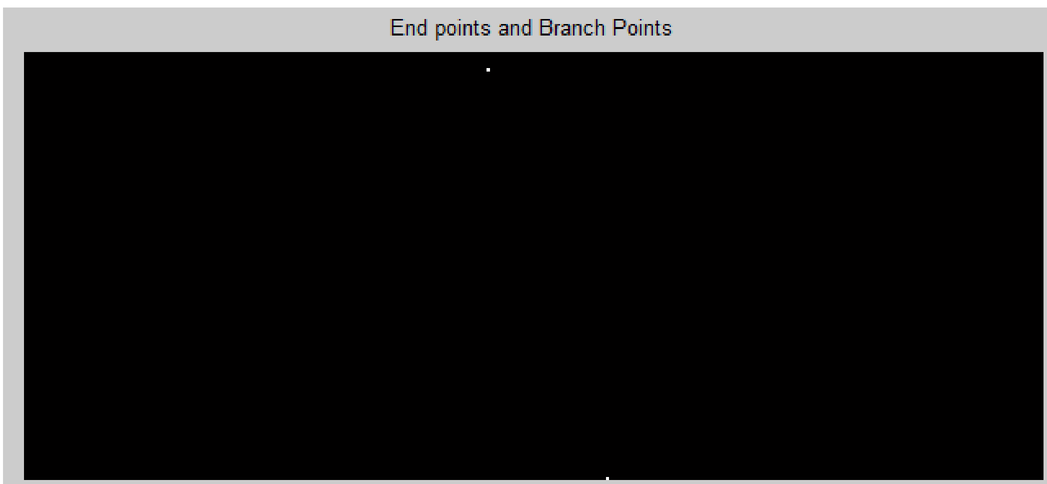

Figure 12: Total number of edge end points

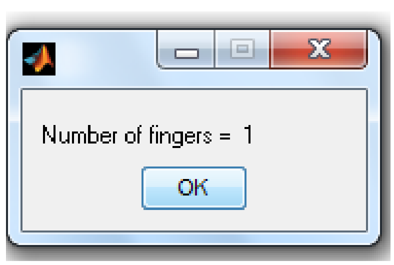

Figure 13: Result

4. RESULTS

The proposed system is tested with different types of hand gesture images taken from 5 subjects .It gives 99% accuracy for 25 different images of hand gestures. It detects the gesture even if fingers are attached to each other. The system gives output in less amount of time. The results obtained by the system are shown in the following charts.

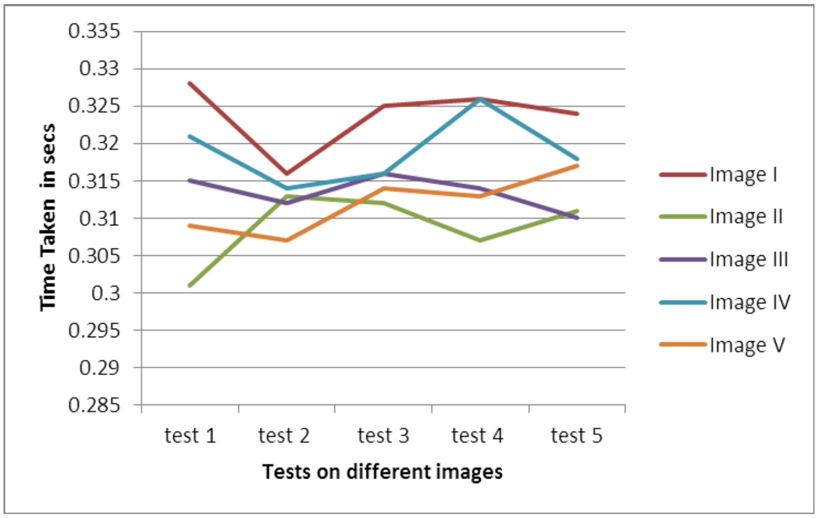

Figure 14: Result of time taken to detect the fingers in Hand Gesture.

The results of our system are compared with other present systems in respect of accuracy and time.

5. APPLICATIONS

The advantages of Accurate End Point Identification (AEPI) Method are that it gives more accurate result in less amount of time. It could be used in various applications like Hand Gesture recognition systems, Automatic Traffic monitoring, Communication systems for disabled peoples etc.

6. CONCLUSION & FUTURE SCOPE

The Accurate End Point Identification method employs various morphological computations which are properly sequenced. The end point detection method accurately identifies the hand gesture from the static input image. Currently the method is tested for the images taken with uniform background and without intervention of other objects. The methods achieved promising results with 99% accuracy as compared to available techniques for hand gesture recognition.

In future, this Accurate End Point Identification (AEPI) Method will be used to recognize hand gestures from images with varying background and having objects other than hand gesture in the image. AEPI method will also be applied for real time hand gesture recognition in future.

REFERENCES

[1] N. Ibraheem, M. Hasan, R. Khan, P. Mishra, (2012). “Comparative study of skin color based segmentation techniques”, Aligarh Muslim University, A.M.U., Aligarh, India.

[2] Serban Oprisescu, Christoph Rasche, Bochao Su, “Automatic Static Hand Gesture Recognition Using Tof Cameras”, 20th European Signal Processing Conference (Eusipco 2012), Bucharest, Romania, August 27 - 31, 2012© EURASIP, 2012 - ISSN 2076-1465.

[3] Ankita Chavda, Sejal Thakkar “A Vision Based Static Hand Gesture Alphabet Recognition” International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 4, April - 2013 ISSN: 2278-0181.

[4] Jayshree R.Pansare, Sandip Kadam, Onkar Dhawade, Pradeep Chauhan, “Real Time Static Hand Gesture Recognition System in Simple Background for Devanagari Number System” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 3, March 2013

[5] Trong-Nguyen Nguyen, Huu-Hung Huynh, “Static Hand Gesture Recognition Using Artificial Neural Network”, Journal of Image and Graphics, Volume 1, No.1, March, 2013.

[6] Gaurav Manik Bidgar, Mangesh Balasaheb Autade “ Hand Gesture Recognition For Hci (Human-Computer Interaction) Using Artificial Neural Network”, International Journal of Technical Research and Applications e-ISSN: 2320-8163, www.ijtra.com Volume 2, Issue 2 (March-April 2014), PP. 48-50

[7] Miss.Neha.B.Bhoyar, Prof. Mahip. Bartere, “Neural Network based Static Hand Gesture Recognition”, International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 2, February 2014 ISSN 2319 – 4847.

[8] Haitham Badi, Sameem Abdul Kareem and Sabah Husien, “Feature Extraction Technique for Static Hand Gesture Recognition”, CHAPTER 2, Editor: A. Chaudhary, Recent Trends in Hand Gesture Recognition DOI: 10.15579/gcsr.vol3.ch2, GCSR Vol. 3, pp. 19-41, 2015© Science Gate Publishing - Available under CC BY-NC 4.0 International License.

[9] Rajashree Patil Shailaja Patil, “Static Hand Gesture Recognition based on DWT Feature Extraction Technique” IJIRST –International Journal for Innovative Research in Science & Technology| Volume 2 | Issue 05 | October 2015 ISSN (online): 2349-6010.

[10] Chethana N S, Divya Prabha, M Z Kurian, “Static Hand Gesture Recognition System For Device Control”, International Journal of Electrical, Electronics and Data Communication, ISSN: 2320-2084 Volume-3, Issue-4, April-2015.

[11] R. P. Mihail, N. Jacobs, and J. Goldsmith “Static Hand Gesture Recognition with 2 Kinect Sensors ,http://weblidi.info.unlp.edu.ar/worldcomp2012-2012-mirror/p2012/IPC4385.pdf

[12] Saxena, A., Jain, D. K., & Singhal, A. (2014, April). Hand Gesture Recognition Using an Android Device. In Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on (pp. 819-822). IEEE.

[13] Elhenawy, I., & Khamiss, A. (2014). The design and implementation of mobile Arabic fingerspelling recognition system. International Journal of Computer Science and Network Security, 14(2), 149-155.

[14] Otsu, N., “A Threshold Selection Method from Gray-Level Histograms,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[15] Soille, P., Morphological Image Analysis: Principles and Applications, Springer-Verlag, 1999, pp. 173-174

[16] Lim, Jae S., Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ, Prentice Hall, 1990, pp. 469-476.

[17] Haralick, Robert M., and Linda G. Shapiro, Computer and Robot Vision, Volume I, Addison-Wesley, 1992, pp. 28-48.

[18] Kong, T. Yung and Azriel Rosenfeld, Topological Algorithms for Digital Image Processing, Elsevier Science, Inc., 1996.

[19] Lam, L., Seong-Whan Lee, and Ching Y. Suen, “Thinning Methodologies-A Comprehensive Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 9, September 1992, page 879, bottom of first column through top of second column.

[20] Pratt, William K., Digital Image Processing, John Wiley & Sons, Inc., 1991.

[21] Eshed Ohn-Bar and Mohan Manubhai “Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations” Ieee Transactions On Intelligent Transportation Systems, Vol. 15, No. 6, December 2014, Pp2368-77.

[22] Carl A. Pickering, Keith J. Burnham, Michael J. Richardson “A Research Study of Hand Gesture Recognition Technologies and Applications for Human Vehicle Interaction” http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.7688&rep=rep1&type=pdf.

[23] Rafiqul Zaman Khan1 and Noor Adnan Ibraheem, “Comparative Study Of Hand Gesture Recognition System”, Natarajan Meghanathan, et. al., (Eds): Sipm, Fcst, Itca, Wse, Acsit, Cs & It 06, Pp. 203–213, 2012. © Cs & It-Cscp 2012 Doi : 10.5121/Csit.2012.2320.

[24] Rafiqul Zaman Khan and Noor Adnan Ibraheem, "Hand Gesture Recognition: A Literature Review", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.4, July 2012

[25] Siddharth S. Rautaray, Anupam Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey", © Springer Science+Business Media Dordrecht 2012.