

# Application Control Using 3D Gesture Recognition

<sup>1</sup>Ashutosh Verule, <sup>2</sup>Shrivardhan Suryawanshi, <sup>3</sup>Poonam Rajput and <sup>4</sup>Rajashri Itkarkar

<sup>1, 2, 3 and 4</sup>Department of Electronics & Telecommunications

Rajarshi Shahu College of Engineering

Affiliated to University of Pune, Pune, Maharashtra, India

<sup>1</sup>[ashutosh.verule@gmail.com](mailto:ashutosh.verule@gmail.com)

<sup>2</sup>[shrieducation1@gmail.com](mailto:shrieducation1@gmail.com)

<sup>3</sup>[poonambrajput@gmail.com](mailto:poonambrajput@gmail.com)

<sup>4</sup>[itkarkarrajashri@yahoo.com](mailto:itkarkarrajashri@yahoo.com)

**Abstract**—A speedy rise in 3D applications and virtual environments in computer systems has led to the need for a new type of interaction device. Our proposed algorithm involves recognition of gestures from their 3D images by their acquisition, segmentation of the gesture information by removing background, extraction of the key features from the gesture using Eigen values and Eigen vectors, determination of the gesture enclosed within the image, and, executing the application as per the gesture identified. Obviously a very efficient method is direct manipulation with bare hands, but, not every human is lucky enough to have hands. This paper shows the possibility to perform non-trivial tasks from the recognized hand gestures using only a few well-known gestures, helping the handicapped negate the dependency on others to go by their mundane routine, selecting any alphanumeric value without any key pressing, replacing the human need of pressing buttons in elevators making them automatic, and other such fledgling applications. This work emphasises on execution of windows based applications, where applications such as MS word, MS excel etc. can be opened using defined gestures.

**Keywords**— Gesture recognition, Eigen value, Eigen vector, 3D gesture, Euclidean distance.

## I. INTRODUCTION

A gesture is some specific motions of body parts that represent a meaningful data. Sign Language is a well-structured code gesture, every gesture having a unique meaning assigned to it. Many research works related to sign languages have been done as for example the American Sign Language, the British Sign Language, the Japanese Sign Language, and so on, which establish standards that are followed to consort a meaning to a particular gesture.

Sign Language is the only means of communication for deaf people. Finding an experienced and qualified interpreter every time is a very difficult task and also unaffordable. Moreover, people who are not deaf, never try to learn the sign language for interacting with the deaf population. This becomes a cause of isolation of the deaf people. But if a system can be programmed in such a way that it can translate sign language to text or audio format, the conversation gap between the normal people and the deaf community can be minimized. We have proposed a system which is able to

recognize the various alphabets and numbers of American Sign Language for Human-Computer Interaction (HCI) giving more accurate results in least possible time.

Not only is gesture recognition limited to the use of deaf and handicapped population, but today's world also has many applications of such systems like robotics, gaming consoles, television control mechanisms and wheel chair automation. With such widespread applications, it is imperative for us to adapt to this changing technology and hence presenting a unique algorithm for gesture recognition from 3D images.

This paper presents our unparalleled algorithm which comprises following major steps:

- 1] Acquiring 3D image of the input gesture through an externally interfaced camera or a readymade database.
- 2] Pre-processing the acquired image to bring it to a standard resolution
- 3] Segmentation of the image to negate the effects of background of the gesture and dynamic lighting conditions in the image.
- 4] Feature extraction: To avoid an erroneous interpretation, even an infinitesimal difference between two gestures must be differentiable. This is done by using covariance matrix, Eigen values and Eigen vectors, covering all the features of the entire image, and, by calculating the Euclidean distance.
- 5] Giving the respective text and audio output, and, executing the application consorted to the identified gesture.

The rest of the paper is organized as follows:

Section II provides a brief review of the varied literature work studied for reference. Section III describes the pertaining theory required for lucid understanding of the algorithm. Section IV explains the proposed system. Section V concludes the work with results and also throws a light towards the future work.

## II. LITERATURE REVIEW

Different approaches have been used by researchers for recognition of hand gestures which were implemented in varied fields. Padmanabham Patki and Nagasrikanth Kallakuri applied the approach of using various algorithms like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Support Vector Machines (SVM) considering linear, polynomial and sigmoid kernels and also Correlation Filter. An accuracy of up to 98.84% was attained using the above mentioned algorithms on a self-generated database.<sup>[6]</sup>

Hazem Khaled *et al.* used the background subtraction technique to extract the ROI (Region Of Interest) of the hand in which the fingertip is detected using logical heuristics equations that are applied on hand contour, convex hull and convexity defects points. This method improves the finger tip's detection by 52%.<sup>[3]</sup> Dharani Mazumdar *et al.* used a new idea called "Finger-Pen", where a glove based system is developed by segmenting only one finger from the hand for proper tracking. Problems such as skin colour detection, complexity from large population in front of the camera, complex background removal and variable lighting condition are found to be efficiently handled by this system.<sup>[2]</sup>

Reza Hassanpour *et al.* carried a detailed survey on the methods of analyzing, modeling and recognizing hand gestures in the context of the HCI.<sup>[1]</sup> Joyeeta Singha *et al.* proposed a system using Eigen value weighted Euclidean distance as a classification technique for recognition of various Sign Languages of India. The system comprises of four parts: Skin Filtering, Hand Cropping, Feature Extraction and Classification which gave a recognition rate of 97%.<sup>[4]</sup> Cristina Manresa *et al.* used a new algorithm to track and recognize hand gestures for interacting with a videogame. This algorithm consists of three main steps: hand segmentation, hand tracking and gesture recognition from hand features. The system's performance is evaluated by showing the usability of the algorithm in a videogame environment.<sup>[5]</sup>

## III. THEORETICAL BACKGROUND

### A. American Sign Language

Sign languages are an ancient form of languages that could be dated back to as early as the advent of the human civilization, when the first theories of sign languages appeared in history. Since then the sign language has evolved and been adopted as an integral part of our day to day communication process. In airports, a predefined set of gestures makes people on the ground able to communicate with the pilots and thereby give directions to the pilots of how to get off on the run-way. In the world of sports too, gestures are common. A single forefinger raised by the umpire in cricket signifies the batsman is out. Furthermore, deaf people have over the years developed a sign language where all defined gestures have an assigned meaning. The language allows them to communicate with each other and the world they live in.

American Sign Language (ASL) is the predominant sign language of deaf communities in the United States and English-speaking parts of Canada. Besides North America, dialects of ASL and ASL-based creoles are used in many countries around the world, including much of West Africa and parts of Southeast Asia. Originated in the early 19th century, ASL use propagated widely via schools for the deaf community organizations.

Figure 1 depicts the gestures representing the numbers from 1 to 10 in American Sign Language, while figure 2 constitutes of all the alphabets in ASL.

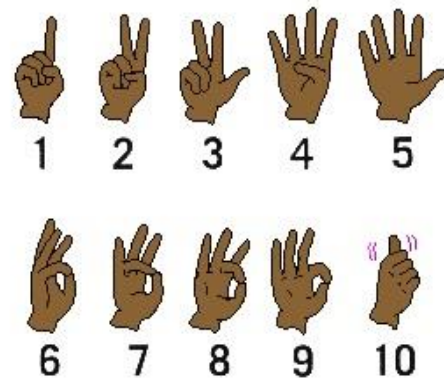


Figure 1: Numbers in American Sign Language

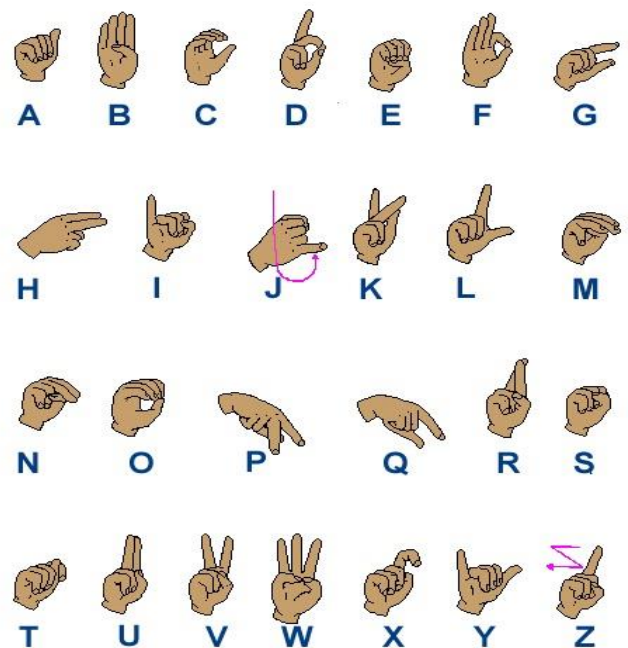


Figure 2: Alphabets in American Sign Language

### B. Eigen Values And Eigen Vectors

An Eigen vector of a square matrix  $A$  is a non-zero vector  $v$  that, when the matrix is multiplied by  $v$ , yields a constant multiple of  $v$ , the multiplier being commonly denoted by  $\lambda$ . That is:

$$Av = \lambda v.$$

The number  $\lambda$  is called the Eigen value of  $A$  corresponding to Eigen vector  $v$ .

Eigen values and Eigen vectors are a part of linear transformations. Eigen vectors are the directions along which the linear transformation acts by stretching, compressing or flipping and Eigen values gives the factor by which the compression or stretching occurs. Eigen vectors are set of basic functions which describes variability of data. And Eigen vectors are also a kind of coordinate system for which the covariance matrix becomes diagonal for which the new coordinate system is uncorrelated. The more the Eigen vectors the better the information obtained from the linear transformation. Eigen values measures the variance of data of new coordinate system.

### IV. PROPOSED SYSTEM

Our proposed system for hand gesture recognition is represented by the block diagram shown in Fig. 3.

The various stages of operation in our system are:

#### 1) Acquiring image:

A three dimensional image of input gesture is at first taken as input through the digital camera. This image is stored in the test database. Also, prior to this, 3D train images of all the gestures are taken by the same camera and stored in the train database for further evaluation.

#### a. Camera specifications:

Name: Cybershot (DSC-HX-20V)

Type: Exmor R CMOS Sensor.

Size: 1 / 2.3 type (7.76mm).

Gross Pixels: Approx. 18.9 Mega Pixels.

Effective Pixels: Approx. 18.2 Mega Pixels.

Lens Type: Sony G Lens

Optical Zoom: 20x

Clear Image Zoom: 40x

#### b. 3D Still Image specifications:

Extension: '.mpo'

18M (4,896 X 3,672) 4:3 mode

13M (4,896 X 2,752) 16:9 mode

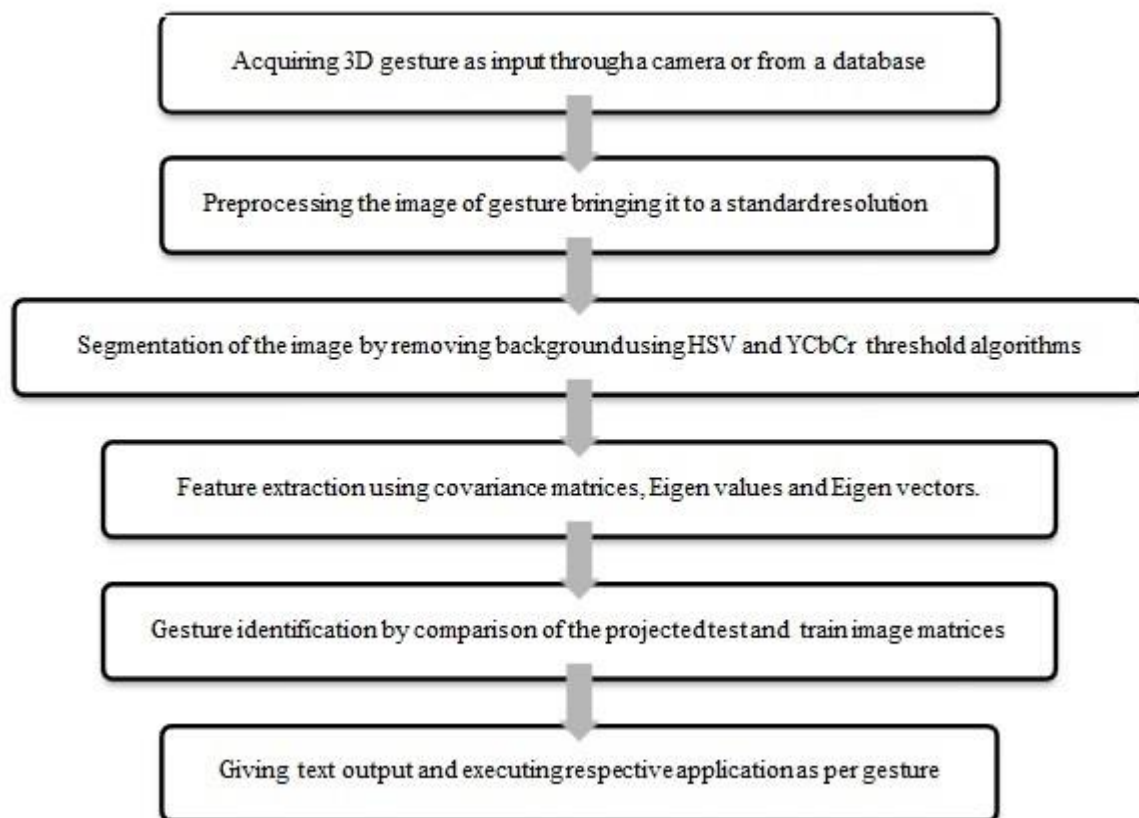


Figure 3: Block Diagram Of The Proposed System

## 2) Pre-processing:

In pre-processing the resolution of the image is reduced i.e. changed as desired, because the resolution of any image for every camera differs. So to obtain this desired format we use the 'imresize' function of MATLAB. We have scaled the image to 0.1, i.e., 10%.

MATLAB version specifications:

Name of the version: R2009B

Version number: 7.9.0.529

64-bit (win64)

## 3) Image segmentation:

This stage mainly does the work of hand detection and background removal from the image. The primary requirement before identifying and classifying the hand signs is to locate the hand in the frame, subtract the background and to be insensitive to lighting conditions. In order to obtain the location of the hand, skin colour recognition was used. The algorithm used for the skin colour identification is to convert the image obtained, which is an image in RGB colour space, to YCbCr model and then set thresholds for the Cb and Cr colour dimensions. The thresholds used are:

$$77 < Cb < 131$$

$$121 < Cr < 160$$

Once we get the pixels that fall within the threshold limits, all the other pixel values are zeroed. Thus a binary CbCr image is obtained from the thresholds.

Although the YCbCr model does detect skin colour effectively from the thresholds, there are certain limitations. The YCbCr model fails when the background comprises of flash/lighting effect, or shadow of the hand used, or when the colour of background is exactly similar to that of skin. Thus, we have used another model called the HSV model along with the YcbCr model. Here, the input RGB image is converted to the HSV image. The motive of performing this step is RGB image is very sensitive to change in illumination condition. The HSV colour space separates three components: Hue which means the set of pure colours within a colour space, Saturation describing the grade of purity of a colour image and Value giving relative lightness or darkness of a colour. Then, thresholds for hue and saturation are used. The brightness component is neglected since we do not want to include the effect of light in background like flash or shadow in our final binary image. The thresholds used for HSV binary image are:

$$0 < H < 0.25$$

$$0.05 < S < 0.9$$

After getting all the components that fall within the threshold limits, all the other pixel values are zeroed. Thus a binary HSV image is obtained from the thresholds.

A logical 'AND' operation is done between the binary HSV and the binary CbCr image to get the most probable skin region (i.e. hand). Noise is minimized using morphological operations like erosion, dilation etc. Fig. 4 shows the processing from the test image to its equivalent binary image after segmentation.



Figure 4: Image Segmentation

Test image followed by YCbCr binary image, HSV binary image and the final binary image (in order from left to right).

## 4) Feature extraction:

After successfully augmenting the hand portion from the image, we must now extract all the features of the image, which are in the form of a matrix. This feature extraction is performed on both the train and test images in the following way.

- The train images are firstly converted to grayscale. Later reshape function is used to form the matrix of the grayscale images, in such a way that each column in the matrix represents the mean values of elements for each and every train image along each of its rows. The matrix will have exactly the same number of columns as the number of train images.
- A covariance matrix of the train images is formed by subtracting the mean value of each row in the above matrix from each element of every image.  

$$\text{CovMatTrainImage}(i,j,k) = \text{TrainImage}(i,j,k) - \text{Mean}$$
- Using this covariance matrix, the Eigen values and Eigen vectors for all the train images are generated. This is done so as to have a detailed extraction of features from the images, without missing out on larger values within the image matrix. An Eigen matrix is then made out of these Eigen values and Eigen vectors. The formula used for this was,

$$Av = \lambda v.$$

The number  $\lambda$  is called the Eigen value of A corresponding to Eigen vector v.







- Based on the Eigen matrix and the covariance matrix, a projected train image matrix is then generated which will have combined results of the covariance matrix and the Eigen matrix. This projected train image matrix will have values of all the train images in each of its columns respectively from which we can display the image represented by that column.
- Similar operations of grayscale conversion, reshape function, covariance and Eigen matrix generation are performed on the test image, so as to get the projected test image matrix representing the input gesture image.

## 5) Projected test and train matrix comparison

Based on the projected test and train image matrices, a Euclidean distance is found as the difference between the average values represented by the elements within them. The column in the projected train image matrix that deviates minimally from the projected test image matrix will have minimum Euclidean distance and vice versa.



Table 1: Results of a few gestures with their Euclidean distances

Test Image	Euclidean Distance with Train image 1	Euclidean Distance with Train image 1	Euclidean Distance with Train image 1	Euclidean Distance with Train image 1	Euclidean Distance with Train image 1	Gesture Recognized	Application opened as per gesture
	2.3576 x e <sup>11</sup>	<b>2.3571 x e<sup>11</sup></b>	2.3698 x e <sup>11</sup>	2.3668 x e <sup>11</sup>	2.3654 x e <sup>11</sup>		MS Excel
	4.0995 x e <sup>11</sup>	6.3094 x e <sup>11</sup>	2.7352 x e <sup>11</sup>	3.5371 x e <sup>11</sup>	<b>1.0453 x e<sup>11</sup></b>		Mozilla Firefox
	<b>0</b>	5.8744 x e <sup>11</sup>	4.7386 x e <sup>11</sup>	5.7671 x e <sup>11</sup>	2.3354 x e <sup>11</sup>		MS Word

Thus, the column with lowest Euclidean distance will represent the identified gesture image amongst all other train gesture images.

$$\text{EucDist} = \text{ProjTestImg}() - \text{ProjTrainImg}().$$

#### 6) Giving text output and opening respective application as per gesture

Based on the identified gesture in the test image, the equivalent numerical and alphabetical output is given out on the screen, for example, an output of '1' and 'One' for the test input as a gesture image representing 1 in ASL.

At the same time, a different Windows application is launched for every gesture identified. For example, if '1' is the identified gesture, MS Word is opened in the system, or if '7' is the identified gesture, MS PowerPoint is opened in the system, and so on.

In this way, using covariance matrix, Eigen values and Eigen vectors, and, Euclidean distance, we can identify any input gesture available in the train database by the above method. Some of the results are shown in the Table 1.

### V. CONCLUSION AND FUTURE SCOPE

A gesture can be identified after the removal of background by finding its covariance matrix. This covariance matrix can then be used to find the Eigen values and Eigen vectors from the matrix for each test and train image. After calculating the Eigen matrices of both test and train image, one can give the desired output declaring the identified gesture, by finding the minimum Euclidean distance between the test and train image, as shown in Table 1.

Although our gesture recognition algorithm is advantageous, it has some limitations. These limitations, listed below, can be worked upon for future work.

- One of the limitations of our algorithm is, if the background happens to be of skin colour, then the YCbCr and HSV results fail to show complete background suppression. Some minor part of 'hand' is lost during segmentation. Techniques like Gaussian filtering can be used to do so<sup>[7]</sup>.
- Furthermore, the execution time for our algorithm is around half a minute per gesture recognized since they are 3D gestures with large file size. Work can also be carried around this to make the algorithm much faster.

### ACKNOWLEDGMENT

We would like to thank the support centre of Mathworks Inc and all its bloggers for their prompt support to answer our questions and doubts unconditionally. We would also like to thank our teacher and guide Prof. Mrs R. R. Itkarkar for her continuous help regarding our paper. Furthermore, we also thank the IEEE associates for providing a large collection of papers based on image processing, which were of great help in studying, analysing and selecting the right path towards our goal in the project.

### REFERENCES

- [1] Reza Hassanpour, Asadollah Shahbahrani, "Human Computer Interaction Using Vision-Based Hand Gesture Recognition", Journal of Advances in Computer Research, 2010.
- [2] Dharani Mazumdar, Anjan Kumar Talukdar & Kandarpa Kumar Sarma, "A coloured fingertip-based tracking method for continuous hand gesture recognition", International Journal of Electronics Signals and Systems (IJESS), ISSN: 2231- 5969, Vol-3, Iss-1, 2013.
- [3] Hazem Khaled, S. Sayed, El Sayed Mostafa ,Hossam Ali, "Hand Gesture Recognition Using Average Background and Logical Heuristic Equations", International Journal Of Computers & Technology, ISSN 2277-3061, Vol. 11 No. 5, 2013.

- [4] Joyeeta Singha, Karen Das, "*Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique*", International Journal of Advanced Computer Science and Applications, Vol. 4, No. 2, 2013.
- [5] Cristina Manresa, Javier Varona, Ramon Mas and Francisco J. Perales, "*Hand Tracking and Gesture Recognition for Human-Computer Interaction*", Electronic Letters on Computer Vision and Image Analysis 5(3):96-104, 2005.
- [6] Padmanabham Patki, Nagasrikanth Kallakuri, "*British Sign Alphabet Recognition System*".
- [7] Aisha Meethian, B.M.Imran, "*Real time gesture recognition using hand tracking system based on GMM*".
- [8] Diedrick Marius, Sumita Pennathur, and Klint Rose, "*Face Detection Using Color Thresholding, and Eigenimage Template Matching*".
- [9] S. Chitra and G. Balakrishnan, "*Comparative Study for Two Color Spaces HSCbCr and YCbCr in Skin Color Detection*". Applied Mathematical Sciences, Vol. 6, 2012, no. 85, 4229 - 4238
- [10] Sofia Tsekeridou and Ioannis Pita, "*Facial Feature Extraction in Frontal Views Using Biometric Analogies*"
- [11] Amanpreet Kaur and B.V Kranthi, "*Comparison between YCbCr Color Space and CIELab Color Space for Skin Color Segmentation*", International Journal of Applied Information Systems (IIAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 3– No.4, July 2012.
- [12] Jorge Alberto Marcial Basilio, Gualberto Aguilar Torres, Gabriel Sánchez Pérez, L. Karina Toscano Medina, Héctor M. Pérez Meana, "*Explicit Image Detection using YCbCr Space Color Model as Skin Detection*", Applications of Mathematics and Computer Engineering, ISBN: 978-960-474-270-7
- [13] Yanjiang Wang, Baozong Yuan, "*A novel approach for human face detection from color images under complex background*", Pattern Recognition 34 (2001) 1983}1992.
- [14] Douglas Chai and King N. Ngan, "*Face Segmentation Using Skin-Color Map in Videophone Applications*", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 9, No. 4, June 1999.
- [15] Rick Kjeldsen and John Kender, "*Finding Skin in Color Images*", 1996 IEEE.
- [16] G. Kukharev and A. Novosielski, "*Visitor Identification Elaborating Real Time Face Recognition System*", Proceedings Of 12<sup>th</sup> Winter School Of Computer Graphics, Plzen, February 2004.