A Static Hand Gesture Recognition System to Recognize the Total number of Fingers

D.K. Vishwakarma* Sahib Majithia† Nikhil Kumar Mishra‡
Department of Electronics and Communication Engineering,
Delhi Technological University
New Delhi, India
Email: *dvishwakarma@gmail.com ,†sahibmajithia@gmail.com, ‡nkmishra1997@gmail.com

***Abstract*—Gesture recognition aims at recognizing meaningful expressions of motion by humans which involve the hands, arms, face, head, or body. It holds great importance in designing an intelligent and efficient human-computer interface. We take into account the two possible fixed geometries to work on. The proposed methods follow the procedure of Preprocessing: which helps with noise removal and image enhancement; Segmentation of hand region: uses skin likelihood method to extract skin color; Feature extraction: Morphological and geometry based functions are used to extract the fingers; and active fingers are counted by method of Rule based Classification. In order to test performance, an experiment is conducted using standard and a self-generated dataset of images. The accuracy achieved on these dataset is greater than on similar state of arts.**

Index Terms*—Hand Gestures, Skin Segmentation, Active Fingers, Finger Count, Mask Generation, Recognition, Low Resolution

I. Introduction

In the present day framework of interactive, intelligent computing, an efficient human computer interaction is assuming utmost importance in our daily lives. Gesture recognition can be termed as an approach in this direction. Gesture recognition is the mathematical interpretation of a human motion by a computing device

There are methods which used additional hardware devices such as data glove devices and color markers to easily extract a comprehensive description of gesture features for recognition [1] . Neural Network classifier has been applied for gestures classification [2] but it is time consuming and when the number of training data increases, the time needed for classification are increased too .Orientation histogram method [3] has some problems which are; different gestures could have similar orientation histograms, and similar gestures might have different orientation histograms. Other methods based on the appearance of the hand use skin color to segment the hand and extract necessary features, these methods are considered easy, natural and less computationally costly compared with methods mentioned before [1] .

Sign Language Recognition, Robot Control automation of home appliances and presentation control are few of many applications of hand gesture recognition systems. Usage of hand gestures can be extended to control real time applications like VLC media player, paint, pdf reader etc. This wireless control of embedded instruments is a great progress towards the field of embedded vision and automation.

This paper proposes two methods for counting the number of active fingers. These are proposed based on the geometric facts of the hand region.

Since the proposed method works on geometric properties and does not require any extracted features; it works well on low-resolution images which is a great advantage for developers regarding data storage.

The algorithm first helps with skin segmentation from the region of interest. Further, the number of active fingers are counted after operating them with a range of morphological operations. In this paper section 2 and 3 describe the algorithms proposed. Further sections discuss the Results (4), Conclusions (5) and Future scope (6) of our work.

II. Hand Segmentation

Skin segmentation obtains the hand region. The skin segmentation has been applied correctly to Region of Interest of hand region. This does not cause susceptibility as the Region of Interest can be easily extracted by different depth estimation strategies, which include:

Active methods: Light-based depth estimation/Pattern projection/Ultrasounds based methods

Passive methods: Monocular depth estimation/Image structure/Points tracking or Optical flow/ Multiview solutions for the depth estimation

The Region of interest can also be extracted by using instruments like Kinect which uses an RGB camera with a depth sensor.

We have used gestures G1, G2, G3, G4 and G5.

(a) G1 (b) G2 (c) G3 (d) G4 (e) G5

Fig. 1: Gestures used with extracted ROI

A. SKIN SEGMENTATION

The most important part of hand gesture recognition is the skin segmentation. Segmentation accuracy determines the eventual success or failure of an automated analysis procedure.For segmentation, we first convert the RGB colored image into the YCbCr color model.

The transformation of the original RGB colour image is done using the following equations:

$$Y = 0.2990R + 0.5870G + 0.1140B \tag{1}$$

$$Cb = -0.1687R - 0.3313G + 0.5000B \tag{2}$$

$$Cr = 0.5000R - 0.4187G - 0.0813B \tag{3}$$

The mean values of chrominance components are used to extract the skin color of the hand by dropping the luminance component because the variation in skin color is more in intensity than in chrominance. In YCbCr color space, the components Cb and Cr represent the chrominance and component Y accounts for the luminance.

In the earlier works, the YCbCr color model has been used by a basic mathematic operation of thresholding on chromaticity and luminance values. No method has yet been settled which determines the thresholding accurately.So, alternatively, we use a modified approach called a skin likelihood model for skin segmentation.

This model is designed from the maximum likelihood criterion, which has the parameters: mean vector and covariance matrix that maximizes the likelihood function. The Gaussian likelihood function [4] has single maxima, and the parameters mean vector (Υ) and covariance matrix (ξ) can be computed as:

$$\Upsilon_i = \frac{1}{n}\sum_{k=1}^{\infty} y_k \tag{4}$$

Where i $\in$ {Cb,Cr} Cb and Cr are the values of chrominance in YCbCr color space.

$$\xi = \frac{1}{n}\sum_{k=1}^{\infty} (y_k - \Upsilon)(y_k - \Upsilon)^T \tag{5}$$

The probability distribution of skin class can be written as:

$$p(\frac{y}{skin}) = g(y, \Upsilon, \xi) \tag{6}$$

$$g(y, \Upsilon, \xi) = \frac{1}{(2\pi)^{\frac{d}{2}}\,(det\,(\xi))^{\frac{1}{2}}} \times exp(-\frac{1}{2}\delta^2) \tag{7}$$

Where d is the dimension of the feature vector, and $\delta^2 = (y - \Upsilon_i)\xi^{-1}(y - \Upsilon_i)^T$ .From equation 4, the pixels can be classified as skin and non-skin color pixels by using the decision boundary. The probability density distribution of the formed Gaussian model is represented in figure: Plotting a Gaussian Curve for this variation yields the Graph in figure 3.

This Method has a great accuracy and calibration advantage over the YCbCr model and is thus more user-friendly. It can be calibrated for all fairer to darker skins with just a change in input skin section (which was trained for the Gaussian); in contrast to change in Cb and Cr values which demand a technical blueprint.

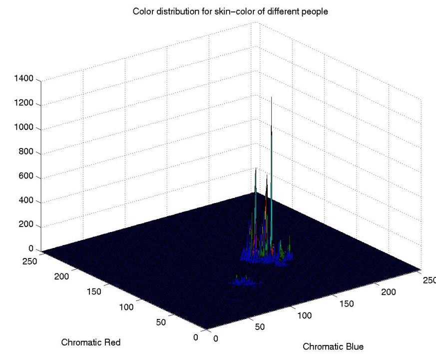


Fig. 2: Graph describing variation of skin tones of human race in Chromatic colour model

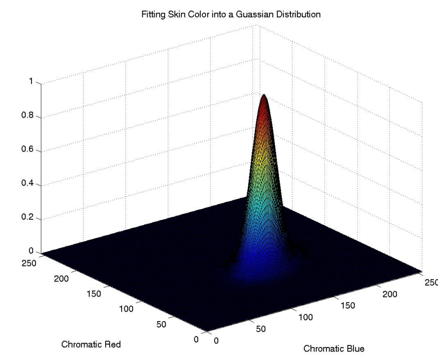


Fig. 3: Gaussian curve of variations in Fig.1

B. PSEUDO CODE:

-imresize([200 200])

SKIN SEGMENTATION:

$trainskinmodel$:

$function\_outputs[mean(cr), mean(cb), cov(cr, cb)]$

$x = [(cr - r_{mean}); (cb - b_{mean})]$;

$likely\_skin(i, j) = [power(2 * pi * power(det(rbcov), 0.5), -1)] * exp(-0.5 * y' * inv(rbcov) * y)$;

For every pixel we calculate :

$likely\_skin(i, j) = [power(2 * pi * power(det(rbcov), 0.5), -1)] * exp(-0.5 * y' * inv(rbcov) * y)$;

%This gives the Gaussian of the image.

$BW = im2bw(likely_skin, thresh)$; %Morphological operations for segmenting the hand shape accurately

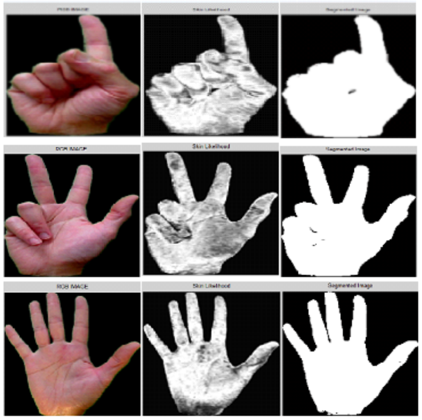
$imclose; imdilate; imfill('holes')$;

Find the hand(palm width)

III. PROPOSED METHODS

A. Method 1: MASKING APPROACH

After acquiring a binary image of the segmented skin region, we extract the count of active fingers by a distinct system of



morphological operations.

1) MORPHOLOGY: Morphology is an extensive collection of image processing procedures that process images based on shapes.

Dilation and Erosion work (at least conceptually) by translating the structuring element to several points in the input image, and analyzing the overlap between the translated kernel coordinates and the input image coordinates. For example, in the case of erosion, the output coordinate set consists of just those points to which the origin of the structuring element can be translated, while the element still remains entirely 'within' the input image.

Mathematically, the main morphological operations we use are denoted as follows:

a) EROSION:

$$(J \ominus K) = [z|(K)_z \subseteq J] \tag{8}$$

b) DILATION:

$$(J \oplus K) = \left[z|(\widehat{K})_z \cap J \neq \phi\right] \tag{9}$$

imopen:

$$(J \ \ o \ \ K) = [(J \ominus K) \oplus K] \tag{10}$$

imclose:

$$(J \cdot K) = [(J \oplus K) \ominus K] \tag{11}$$

2) OPERATIONS: The palm width calculation is needed for obtaining the finger width.

The finger width lies between 1/3 to 1/4 times the palm width depending on the stretch of fingers.

An erosion filter is then designed which is of the dimension of the finger girth. This filter is employed for erosion on the binary segmented image. Dilation of this image helps obtain the mask (here palm).

Morphological opening, closing, and other fitting operations should be executed to accomplish a noise-free and an explicitly shaped mask.

The subtraction of the mask [5] from the binary image provides us the active fingers of the image. Further thresholding for the area of the acquired portions presents us with the precise count of active fingers.

3) PSEUDO CODE: **Masking Approach**

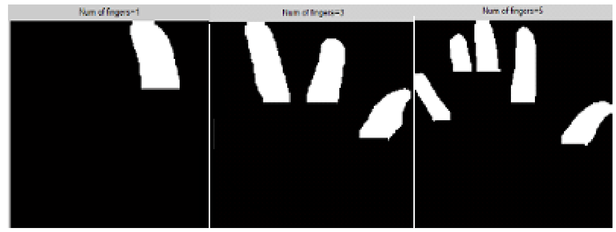
3 <ratio <4

$erode\_filter = ones(1, floor(width/ratio)$; % remove small components such as noises and uneroded finger parts

$bwareaopen(eroded_image, 40)$; %dilate to obtain the original palm shape and size.

$fingers = palm. * BW$

$regionprops \rightarrow$ Area threshholding to count the no. of bounding box to count the number of fingers.



B. Method 2: GRADIENT APPROACH

After acquiring a binary image of the segmented skin region, we extract the count of active fingers by using the information of change of gradients at the finger-tip region. Once the centroid of the binary image is obtained, the shortest distance of the boundary of the silhouette from this centroid point is determined. This minimum distance is determined in all directions from the centroid and the area thus formed depicts the mask for the binary silhouette. This area usually comprises the palm region of the hand and hence forms the base of the mask [6]. Mask subtraction then gives us the fingertip section in addition to the wrist area.

Gradients are calculated by using filters :

Gx: forward horizontal difference

Gy: forward vertical difference

$$Gradient = \sqrt{G_x^2 + G_y^2} \tag{12}$$

We use sobel gradient operator for our method.

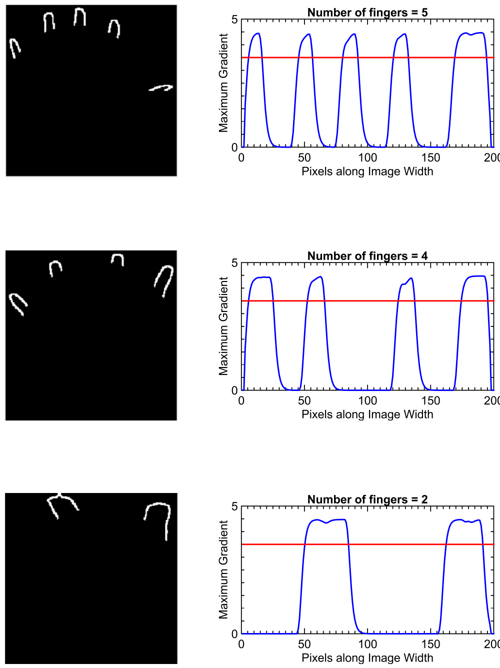
$$\begin{array}{cc} G\text{x} & G\text{y} \\ \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} & \begin{bmatrix} -1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \end{array}$$

Gradient map of the obtained image is calculated. The geometry of the hand infers that the wrist area has a small change of slope at the boundary region though the change is high for the fingertips.

Thus a suitable threshold is established to segregate the wrist area.

The gradient map is passed through a low-pass filter to remove the noise effects. The low-pass filter also helps identify the false high-gradient-change locations.

The final gradient map is a plot for column locations, and thresholding identifies the count of fingers.



1) PSEUDO CODE: **Gradient Approach**

$erode\_filter = strel('disk', 5)$; % remove small components such as noises and uneroded finger parts

$[Gmag, Gdir] = imgradient(erodedimage)$;

Find the change and direction of gradient at the boundary of the entire hand region.

$pdist$(co-ordinates of each boundary point and centroid ,'euclidean');

Find the Euclidean Distance between every boundary point from the center.

Low pass filter $\rightarrow$ Removes the noise effects, identifies the false high-gradient-change locations.

Plot the final gradient map $\rightarrow$ column locations, and thresholding identifies the count of fingers.

IV. RESULT

The rule-based classifiers learned model is represented as a set of if-then rules. Let the correct number of tests be c out of a total of t tests. Accuracy is then given by $(c/t) * 100$.

A. STANDARD DATA-SET

(a) G1 (b) G2 (c) G3 (d) G4 (e) G5

Fig. 4: Hand Gestures used from Standard Database

We applied our algorithms on standard Massey University database [7] of images available online.We used only 5 hand gestures from the data-set

Accuracy by Method 1: - 93.622%
Accuracy by Method 2: - 90.136%

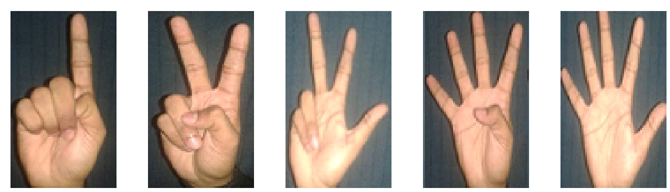
TABLE I: Confusion Matrix for Masking Approach

	Predicted Class				
Actual Class	G1	G2	G3	G4	G5
G1	0.9275	0.0725	0	0	0
G2	0	1	0	0	0
G3	0	0	0.92	0.08	0
G4	0	0	0.014	0.972	0.014
G5	0	0	0	0.0923	0.8616

TABLE II: Confusion Matrix for Gradient Approach

	Predicted Class				
Actual Class	G1	G2	G3	G4	G5
G1	0.884	0.116	0	0	0
G2	0.0143	0.9429	0.0428	0	0
G3	0.0133	0.12	0.8534	0.0133	0
G4	0.014	0.014	0.07	0.888	0.014
G5	0	0	0	0.0615	0.9385

B. OUR DATA-SET



(a) G1 (b) G2 (c) G3 (d) G4 (e) G5

Fig. 5: Hand Gestures used from our created dataset

To have an understanding of real time results, we collect our own database of images and further test the accuracy. We collected hand gestures of different Indian persons.

Accuracy by Method 1 : - 96%
Accuracy by Method 2 : - 98%

TABLE III: Confusion Matrix for Masking Approach

	Predicted Class				
Actual Class	G1	G2	G3	G4	G5
G1	1	0	0	0	0
G2	0	0.9	0.1	0	0
G3	0	0	0.1	0	0
G4	0	0	0	0.1	0
G5	0	0	0	0.3	0.7

V. CONCLUSION AND FUTURE SCOPE

This paper deals with two algorithms in order to recognize the hand gestures and also compares their efficiencies with each other as well as established methods [10]. In our work, the geometry of the hand has been used for the recognition of hand gesture. It is a traditional type of algorithm and is

TABLE IV: Confusion Matrix for Gradient Approach

Actual Class	Predicted Class				
	G1	G2	G3	G4	G5
G1	0.1	0	0	0	0
G2	0	0.1	0	0	0
G3	0	0	0.1	0	0
G4	0	0	0	0.1	0
G5	0	0	0	0.3	0.7

TABLE V: Comparison of various methods

Method for feature extraction and classifier used	Average Accuracy
Bayes classifier [8]	83.40%
PCA	72.73%
Kumar's Autoencoders [9]	83.36%
Masking Approach	93.62%
Gradient Approach	90.13%

thus robust.The proposed methods do not require any training phase [11] to identify the hand gestures so is better than neural networks.

Three dimensional hand model based approaches [12] are gaining interest in the field of Gesture Recognition. The polpular gesture sensors used presesntly are:

A. Time-of-Flight Sensors: ToF-based 3D hand gesture recognition system uses point gestures to control a robot [13].

B. Kinect

1) skeletonbased recognition [14] and

2) depth-based recognition.

C. Leap Motion : Leap motion is a newly released sensor that focuses on the accurate 3D hand positioning. It focuses on a specific area and has gained popularity over Kinect sensor in terms of better accuracy [15].

REFERENCES

[1] S. G. Wysoski, M. V. Lamar, S. Kuroyanagi, and A. Iwata, "A rotation invariant approach on static-gesture recognition using boundary histograms and neural networks," in *Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on*, vol. 4, Nov 2002, pp. 2137–2141 vol.4.

[2] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '91. New York, NY, USA: ACM, 1991, pp. 237–242. [Online]. Available: http://doi.acm.org/10.1145/108844.108900

[3] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *International workshop on automatic face and gesture recognition*, vol. 12, 1995, pp. 296–301.

[4] D. Vishwakarma and R. Kapoor, "An efficient interpretation of hand gestures to control smart interactive television," *Int. J. Comput. Vis. Robot*, pp. 1–18, 2015.

[5] D. K. Vishwakarma, R. Maheshwari, and R. Kapoor, "An efficient approach for the recognition of hand gestures from very low resolution images," in *2015 Fifth International Conference on Communication Systems and Network Technologies*, April 2015, pp. 467–471.

[6] D. K. Vishwakarma and R. Kapoor, "Simple and intelligent system to recognize the expression of speech-disabled person," in *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, Dec 2012, pp. 1–6.

[7] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2d static hand gesture colour image dataset for asl gestures," 2011.

[8] M. Avraam, "Static gesture recognition combining graph and appearance features," 2014.

[9] V. Kumar, G. C. Nandi, and R. Kala, "Static hand gesture recognition using stacked denoising sparse autoencoders," in *Contemporary Computing (IC3), 2014 Seventh International Conference on*. IEEE, 2014, pp. 99–104.

[10] R. Z. Khan and N. A. Ibraheem, "Hand gesture recognition: a literature review," *International journal of artificial Intelligence & Applications*, vol. 3, no. 4, p. 161, 2012.

[11] D. K. Vishwakarma, Priyadarshani, and K. Singh, "A framework for recognition of hand gesture in static postures," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, April 2016, pp. 294–298.

[12] H. Cheng, L. Yang, and Z. Liu, "Survey on 3d hand gesture recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1659–1673, 2016.

[13] D. Droeschel, J. Stückler, and S. Behnke, "Learning to interpret pointing gestures with a time-of-flight camera," in *Proceedings of the 6th international conference on Human-robot interaction*. ACM, 2011, pp. 481–488.

[14] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[15] H. Regenbrecht, J. Collins, and S. Hoermann, "A leap-supported, hybrid ar interface approach," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*. ACM, 2013, pp. 281–284.