International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-5,May 2016
ISSN: 2395-3470
www.ijseas.com

Gesture Recognition Using Full Privacy

Manu Ramachandran[1], Megha Mohankumar[2], Amrutha Nair[3], Nikhil Garad[4]
Guided By: Prof. Padma Nimbhore
[1,2,3,4] Department of Computer Engineering, MIT Academy Of Engineering, Pune, India

Abstract
Systems enabled with Human Interaction is the latest trend in the market, considering the growing requirements, security threats and need of better interfaces. Gestures can be considered as the perfect example for carrying out this trend. Gestures enable users to have an interaction with the system in a user-friendly, cost-effective and time efficient manner. This technology finds use in a wide range of fields. These include Gaming, Object/Motion Tracking, and System Application Control etc. The proposed system highlights a few applications of gesture recognition, with main emphasis on sign language recognition using gestures. This application is designed to use the integrated webcam of any computer, if provided, else an external webcam is used. The webcam captures live video frames. These are then processed to identify gestures for its corresponding translation. We use Hand gestures as inputs for gestures. These gestures are then mapped to its corresponding translation. The system's accuracy can be improved by using webcams of higher resolutions. Gestures Recognition System Using Full Privacy aims to demonstrate the use of gestures for recognizing Sign Language made by deaf and dumb people by means of Hand movement. As the name suggests, a level of privacy is also maintained in the system. The system provides a way in which users can interact with the system easily in a user-friendly manner to recognize the gestures made by disabled people.
Keywords: *Gesture Recognition System, Hand gestures, Human Interaction, Sign Language.*

1. Introduction

Gestures [10] have been used as an alternative form to communicate with computers in an easy way. This kind of human-machine interfaces would allow a user to control a wide variety of devices through hand gestures. Human gestures can be a form of a non-verbal interaction among 2 or more humans. The system thus created for the gesture recognition uses different algorithms to identify gestures in the most efficient way. There are a lot of devices which are applied to sense body position and orientation, facial expression and other aspects of human behavior or state which can be used to determine the communication between the human and the environment. [6] The interfacing device thus created in a collaboration of the human body movements and the sensing device is used to sense them.

For a very long time, many technologies have been developed for both hand and body gesture recognition. The task of recognition is highly challenging because human body is highly complex in terms of its minute movements. Most work in this research field tries to elude the problem by using markers, marked gloves or requiring a simple background. Lately with the advancement of technology, automatic gesture recognition came into trend. These techniques do not require the user to wear extra sensors, clothing or equipment for the recognition system. [6]

The basic idea is to design a real time hand movement detection and a corresponding gesture recognition system. To support this system, the human body positions and movements according to a center point must be traced and interpreted in order to recognize the meaningful gestures.

We present a gesture recognition system that can be an aid for communicating with hearing and speech disables people. In this system, we are taking the input from a simple camera and necessary processing steps are done. These gestures are then compared with the ones stored in the database, which are mostly universal recognized sign languages. The translation is then provided to the user in the form of text or speech.

2. Literature Survey

Kouichi et.al. [1] initially proposed a gesture recognition system that identified Japanese sign language. Static postures like finger alphabets (42 symbols) were taken as inputs that were processed using Neural Networks (NNs) with a three layered

back propagation algorithm. Apart from this, he also developed a Sign Language Word Recognition system applying recurrent neural networks in order to deal with dynamic processes that required data gloves.

Michal R. and William F. [2] have applied a simple pattern recognition technique to the problem of hand gesture recognition. For static hand gestures, they use the histogram of local orientations as a feature vector for recognition. The input gesture's histogram orientation is created which is compared with the stored orientations to perform the corresponding action. The technique works well to identify hand gestures from a training vocabulary of gestures for close-up images of the hand.

Tin H. [3] developed a gesture recognition system that efficiently recognizes Myanmar Alphabet Languages gestures using Neural Networks (NN). His proposed system works autonomously with efficiency for real time gestures which further provides ease in usage. There are no limitations such as usage of gloves or any requirement like uniform background. He used pattern recognition of oriented histograms that proved to be efficient for small data sets.

Xingyan L. In. [4] describes the implementation and testing of a static hand gesture recognition system using FCM. The Fuzzy C-Means algorithm provided enough speed and sufficient reliability to perform the desired task. From the experiments, we also can conclude that the training data should consist primarily of good examples also the recognition accuracy drops quickly when the distance between the user and the camera is greater than 1.5 meters or when the lighting is too strong.

Keskiin C. et.al. [5] Developed a model for gesture recognition, wherein they worked on real time 3D hand gesture recognition by acquiring hand gestures of the user wearing a colored glove, where the hand coordinates are obtained via 3D reconstruction from stereo. The markers in bitmap images and processing is done using various image processing algorithms. Next they have used trained HMMs to recognize different gestures using the input sequences. In their system, they have done recognition of eight defined gestures.

3. Methodology

The whole process of recognizing and interpreting human gestures is the major challenge that is faced. The solution to this can be presented in the form of a strategy, devised to combat the problems that appear throughout the process. [13]

3.1 Skin Detection

After selecting the particular region, the next step involves recognizing the skin region of the user, so as to recognize the gesture made. This is done by detecting the skin color from the background. This involves using various models like hue-saturation-value (HSV), hue-saturation-lightness (HSL) and hue-saturation-intensity (HSI).
[7] A skin detector involves collecting skin patches from different images, choosing a suitable color space from the above mentioned and defining parameters for the classifier.
Skin color detection is tedious as it depends on varying lighting conditions. It is vulnerable and is highly probable of generating false results, owing to change in skin color in different lighting conditions. In this case, the lighting has been taken as invariant in most of the cases, so as to create a stable image for analysis and detection.
A HSV color space model and a YCbCb color model has been used in this scenario. The same function has been overridden to provide detection of skin parts from the real-time image.

3.1.1 Color Spaces

1. RGB Model
   The RGB is a basic color model depicting the color values of the Red, Green and Blue components. It is not considered for color detection because of chrominance and luminance information and non-uniform characteristics.
   Hence it is preferred to use the RGB value by converting into suitable color space models for detection. [7]

2. HSV Model
   [7] Conversion of RGB to HSV model implies the discrimination of color and intensity. Even though useful, it is a time-

taking and expensive transformation. The conversion can be depicted as:

$$H = arccos \frac{\frac{1}{2}(2R - G - B)}{\sqrt{(R-G)^2 - (R-B)(G-B)}} \quad (1)$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} \quad (2)$$

$$V = \max(R,G,B) \quad (3)$$

3. YCbCr Model

[7] It is much easier to separate out color and intensity information while conversion to YCbCr model as compared to HSV. The component Y depicts luminance information. Color information is stored as two components which is the difference between the color and a reference value. Cb is the difference between blue component and a reference value, and Cr is the difference between red component and a reference value.

The conversion of RGB color space to YCrCb color space can be depicted as:

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.279 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

3.1.2 Skin Detection Algorithm

The steps involved in skin detection is explained as follows:

3.1.2.1 Using HSV Model

1. Input image is obtained from the real time video captured by the camera
2. This image, which is in the RGB color model is converted to HSV color space using the aforementioned formulas.
3. Out of the pixels, the skin pixels are selected for masking, based on the threshold value set for the skin color range.
4. The output image contains only skin colored pixels.

3.1.2.2 Using YCbCr Model

The conversion of an RGB image into a YCbCr format results in the resultant image to be comprised of three components, luminance component Y, and chrominance components Cb and Cr. The threshold value of the chrominance components are given as 100<Cb<150 and 150<Cr<200.

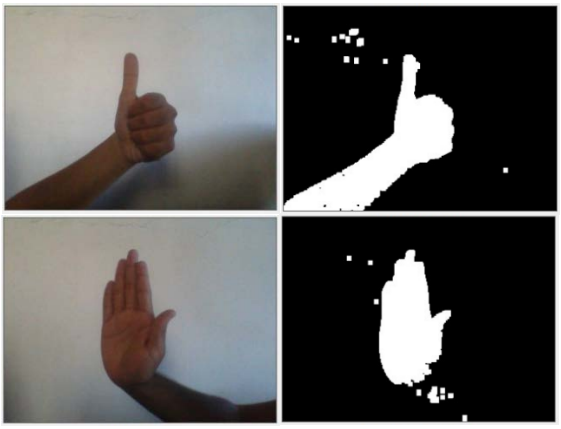


Figure 1: Skin Detection of Hand gestures Using YCbCr model

3.2 Blob Analysis

It is a computer vision based technique used to analyze and detect consistent image regions. During skin detection, skin regions are separated distinctly from the background, providing the fact that the skin color is different from the background. This is achieved with help of the adaptive skin detection model used.

To detect the gestures, the skin region is separated out providing knowledge of the hand position and posture. The skin pixels are then combined to form a consistent region, called a blob. The pixels coming within the blob are considered to be skin pixels, whereas the outer parts are considered as false cases. [8]

A skin segmentation probability is mentioned so as to define which region in the image is considered a blob. This is done by binarizing the image with a skin thresholding value of $P_{th} = 0.74$ to find blobs. Doing so indicates that if a specific region contains more than 74% of skin pixels, it is considered a blob. Connected skin pixels are grouped on the above mentioned conditions to form a group of white pixels, calling it a binary blob. Amongst these binary blobs, the centers between the blobs are calculated and closer blobs are merged together based on the minimum distance between the blobs.

To eliminate false cases, a threshold image size is also provided to detect the skin regions. If a region has an area of more than 200 sq. pixels, it is considered to be a blob.
Now the blob origins detected are further used to detect a complete blob. This is done because in the selected region, though there is a higher ration of skin pixels, it also contains non-skin pixels. The blob is filled with the 4-connectivity model using flood-fill algorithm.

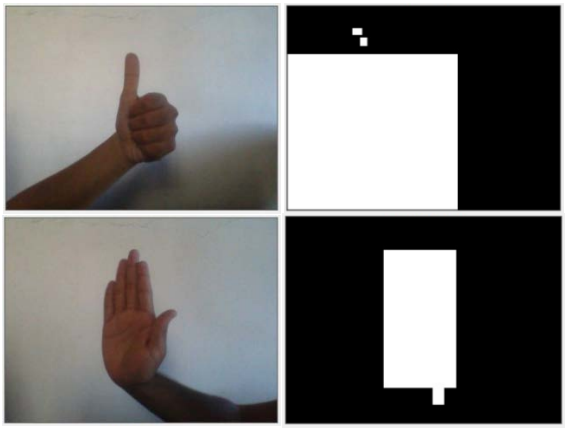


Figure 2: Blob detection of hand parts based on skin-colored pixels

3.3 Template Matching

Template matching is a technique that allows identification of parts of an image that match with the template. This requires two primary components:

1. *Source Image(I):* It is the image in which we expect to find a match to the template.
2. *Template Image(T):* It is the patch image which is compared to the template image stored in the database.

Template matching follows two basic approaches, a feature-based approach and an area based approach.

1. *Feature-based approach:*
   The approach here is to provide a pair-wise match of the template and the reference image. This is done via description of features like spatial relations, wavelets, etc.
2. *Area-based approach:*
   It involves feature detection and feature matching and is mostly used for motion tracking and handling.

Similarity between images is done by computing a measure of both the images in an equal dimension. The similarity is attained by using image correlation, which is achieved by the cross correlation technique. Cross correlation is a sum of pairwise multiplications of corresponding pixel values of the images. The equation for cross correlation is specified for two images t_1 and t_2, and (x, y) as:

$$C = \sum_{x,y} t_1(x,y) t_2(x,y) \qquad (5)$$

Since cross correlation is a value that is variant to position and image intensity, the normalized cross-correlation is used. The correlation is normalized for the effects of intensity and template size changes.

$$C = \frac{\sum_{x,y}[t_1(x,y) - \bar{t_1}][t_2(x,y) - \bar{t_2}]}{\sqrt{\left(\sum_{x,y}[t_1(x,y) - \bar{t_1}]^2 \sum_{x,y}[t_2(x,y) - \bar{t_2}]^2\right)}} (6)$$

This returns a value in the range of 0 to 1. Accepted matches are specified using a threshold value.

In this condition, it is rare to get a perfect match for the templates with the image that is captured in the video frame. Hence, Eigen spaces are used. Eigen spaces provide matching of images in uneven illumination, contrasts and similar hand positions that can be accepted [9].
Eigen object recognizer is used wherein we have a set of training images, i.e. our template datasets in the application, and a new image is provided for recognition based on the set of images. The match between the source image against the data set is done using the normalized correlation equation mentioned above.

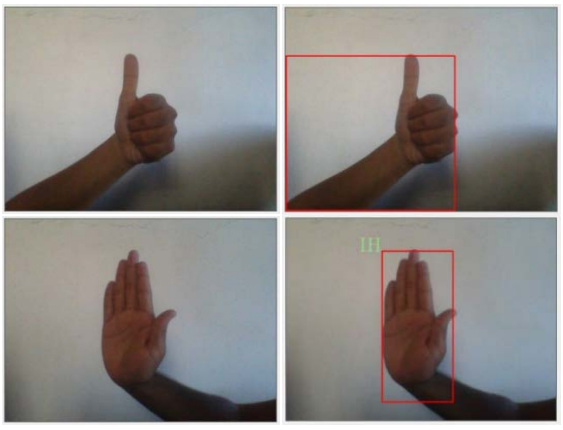
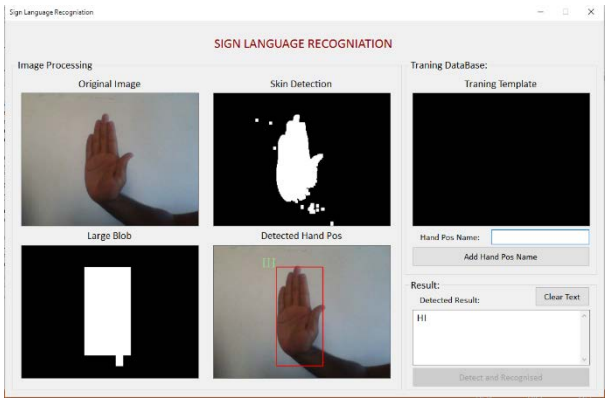


Figure 3: Matching of gestures with the datasets that the system is trained with

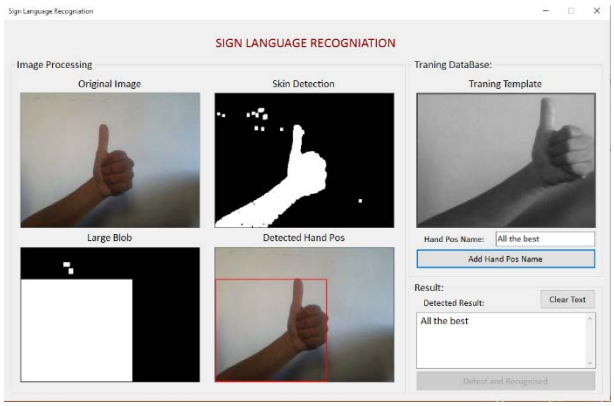
4. Results



After performing the above explained method on given input, the following output was obtained:

Figure 4: Recognition Phase

According to the algorithm, after the above gesture (Picture box 1) is given as an input to the webcam, the image is captured and its corresponding skin detected area is obtained (Picture box 2). Using this, a blob of connected skin colored pixels is obtained (Picture box 3). The red bounded box (Picture box 4) denotes the gesture that is to be saved in the dataset which can be used for comparison later. These are the steps in the



training phase.

In the recognition phase, the input gesture undergoes the above said processes and then compares with the elements in the dataset. When a match is obtained, the matched image is displayed (Picture box 5) and its corresponding meaning is displayed in the text box.

Below is the result for another gesture:

Figure 5: Training Phase

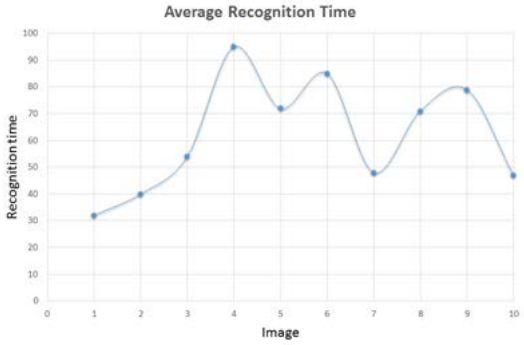


Figure 7: Graph for Recognition Rate

Thus, different gestures were trained along with their corresponding meanings in the training phase and appropriate results were obtained in the recognition phase.

4.1 Recognition Rate

The above figure is a graphical representation of the average time taken to identify different gestures already stored in the dataset accurately. The x-coordinate shows the image stored in the dataset whereas the y-coordinate represents the time to recognize the gestures in milliseconds.
Initially, we created a dataset wherein 5 gestures were added in the training phase. Then, in the recognition phase, time taken for each gesture to give its corresponding translation was noted down for 3 different runs and its average was considered. Thus, this way the average time taken for each gesture was taken. After that, another set of 5 gestures were added and their average recognition time for 3 runs were noted down. Hence, these average values were then plotted to obtain the above graph.

4.2 Efficiency

Also, under uniform lighting conditions and background, we achieve a recognition rate of 90% and are still working to solve the lighting problems to achieve better efficiency.

4.3 Comparison Data

We have compared the parameters of our project with two similar projects. The results are shown below in a tabular form.

Table 1: Comparison with other algorithms and systems

	Xingyan	Tin H	Our work
Input device	Digital Camera	Digital Camera	Web cam
Segmentation	Threshold	Threshold	Threshold
Feature Representation	One dimensional array of 13 element	Orientation Histogram	Blob analysis + skin detection
Algorithm	Fuzzy C-means	Supervised Neural Network	Template Matching
Lighting Conditions	Uniform lighting	Uniform lighting	Moderately uniform lighting
Time Complexity	Moderate but increases with number of elements	Moderate but high in training phase	Less
Recognition rate	85.83%	90%	90%

5. Conclusions

After browsing through all the existing methodologies till date, it has been observed that there are various merits and demerits of the method used for recognition. The proposed system aims at designing a sign language recognition system that incorporates the integrated webcam of a system. The basic idea still remains identifying the dynamic hand gesture input and processing it to match with the database to find translation.

Acknowledgments

We would like to express our sincere gratitude for the assistance and support provided by Prof. Padma Nimbhore, Department of Computer Engineering, MIT Academy of Engineering. We would like to thank her for her valuable guidance that she has provided us at various stages throughout the project work. She has been a great source of motivation enabling us to give our best efforts in this project. We are also grateful to Prof. Uma Nagaraj, Head of Computer Department, MIT Academy of Engineering for encouraging us and rendering all possible help, assistance and facilities.

References

[1] Kouichi Murakami and Hitomi Taguchi, “Gesture recognition using Recurrent Neural Networks,” ACM, pp. 237-242., 1999.

[2] William T. Freeman and Michal Roth, 1994. “Orientation Histograms for Hand Gesture Recognition”, IEEE International Workshop on Automatic Face and Gesture Recognition, Zurich.

[3] Tin Hninn H. Maung, 2009. “Real-time hand tracking and gesture recognition system using neural networks”, World Academy of Science, Engineering and Technology 50, pp. 466- 470.

[4] Xingyan Li, 2003. “Gesture recognition based on fuzzy C-Means clustering algorithm”, Department of Computer Science. The University of Tennessee. Knoxville.

[5] C. Keskin, A. Erkan, L. Akarun, 2003. “Real time hand tracking and 3D gesture recognition for interactive interfaces using HMM”, In Proceedings of International Conference on Artificial Neural Networks.

[6] https://en.wikipedia.org/wiki/Gesture_recogniti on

[7] Khamar Basha Shaik et.al., “Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space”, 3rd International Conference on Recent trends in Computing 2015.

[8] Michal Kawulok, “Adaptive Skin Detector enhanced with blob analysis for gesture recognition”, Institute of Computer Science, Silesian University of Technology, Poland.

[9] Nazil Perveen et.al., “An overview of Template Matching Methodologies and its Applications”, International Journal of Research in Computer and Communication Technology, Vol 2, Issue 10, October 2013

[10] Manu Ramachandran, et.al, ”Gesture Recognition using Full Privacy”, International Journal of Advance Foundation and Research in Computer, Vol 3,Issue 1-Jan 2016.