

September 2011

Hand Gesture Recognition Using Different Algorithms Based on Artificial Neural Network

Shweta K. Yewale

Prof. Ram Meghe Institute of Technology & Research, Badnera,, shwetayewale@rediffmail.com

Pankaj K. Bharne

Sipna College of Engg. & Technology, Amravati, pankajbharne@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijess>



Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Yewale, Shweta K. and Bharne, Pankaj K. (2011) "Hand Gesture Recognition Using Different Algorithms Based on Artificial Neural Network," *International Journal of Electronics Signals and Systems*: Vol. 1 : Iss. 2 , Article 11.

DOI: 10.47893/IJESS.2011.1025

Available at: <https://www.interscience.in/ijess/vol1/iss2/11>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Electronics Signals and Systems by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.



Hand Gesture Recognition Using Different Algorithms Based on Artificial Neural Network



¹Shweta. K. Yewale, ²Pankaj. K. Bharne

Prof. Ram Meghe Institute of Technology & Research, Badnera,

²Sipna College of Engg. & Technology, Amravati

Email: shwetayewale@rediffmail.com¹, pankajbharne@gmail.com²

Abstract - Gesture is one of the most natural and expressive ways of communications between human and computer in a real system. We naturally use various gestures to express our own intentions in everyday life. Hand gesture is one of the important methods of non-verbal communication for human beings. Hand gesture recognition based man-machine interface is being developed vigorously in recent years.

This paper gives an overview of different methods for recognizing the hand gestures using MATLAB. It also gives the working details of recognition process using Edge detection and Skin detection algorithms.

Keywords— *Gesture Recognition, Hand Gesture Recognition, Edge Detection, Artificial Neural Network.*

I. INTRODUCTION

Human hand gestures provide the most important means for non-verbal interaction among people. They range from simple manipulative gestures that are used to point at and move objects around to more complex communicative ones that express our feelings and allow us to communicate with others.

Hand gesture recognition based man-machine interface is being developed vigorously in recent years. Due to the effect of lighting and complex background, most visual hand gesture recognition systems work only under restricted environment.

Many methods for hand gesture recognition using visual analysis have been proposed for hand gesture recognition. Sebastian Marcel, Oliver Bernier, Jean Emmanuel Viallet and Danieal Collobert have proposed the same using Input-output Hidden Markov Models [1]. Xia Liu and Kikuo Fujimura have proposed the hand gesture recognition using depth data [2]. For hand detection, many approaches use color or motion information [3, 4]. Attila Licsar and Tamas Sziranyi have developed a hand gesture recognition system based on the shape analysis of the static gesture [5]. Another method is proposed by E. Stergiopoulou and N. Papamarkos [6] which says that detection of the hand region can be achieved through color segmentation. Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang and Toskiaki Ejima have suggested the method of Hand Gesture Recognition using Hidden Markov models [7]. Another very important method is suggested by Meide Zhao, Francis K.H. Quek and Xindong Wu [8].

They have used AQ Family Algorithms and R-MINI Algorithms for the detection of Hand Gestures.

There is another efficient technique which uses Fast Multi-Scale Analysis for the recognition of hand gestures as suggested by Yikai Fang, Jian Cheng, Kongqiao Wang and Hanqing Lu [9], but this method is computationally expensive. Chris Joslin et. al. have suggested the method for enabling dynamic gesture recognition for hand gestures [10]. Rotation Invariant method is widely used for texture classification and recognition. Timi Ojala et. al. have suggested the method for texture classification using Local Binary Patterns [11].

II. GESTURE RECOGNITION

Gestures are expressive, meaningful body motions – i.e., physical movements of the fingers, hands, arms, head, face, or body with the intent to convey information or interact with the environment. [12]

There are several aspects of a gesture that may be relevant and therefore may need to be represented explicitly. Hummels and Stappers (1998) [2] describe four aspects of a gesture which may be important to its meaning:

- Spatial information – where it occurs, locations a gesture refers to.
- Pathic information – the path that a gesture takes.
- Symbolic information – the sign that a gesture makes.

- Affective information – the emotional quality of a gesture.

In order to infer these aspects of gesture, human position, configuration, and movement must be sensed.

Gesture recognition is the process by which gestures made by the user are made known to the system. [13] Gesture recognition is also important for developing alternative human-computer interaction modalities [14]. It enables human to interface with machine in a more natural way.

Gesture recognition is a technique which used to make computers ‘see’ and interpret intelligently is becoming increasingly popular. Dynamic gesture recognition isn’t something entirely new. [15]

III. ALGORITHMS FOR HAND GESTURE RECOGNITION

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB provides the Toolboxes that allow you to learn and apply specialized technology.

The recognition of hand gesture using MATLAB is described as follows.

We use two algorithms for hand gesture recognition using MATLAB as Edge detection and Skin detection algorithms.

A. EDGE DETECTION

Following steps are used for detecting the edges:

1. Image capturing using a webcam or the front camera of the mobile phone.
2. Converting the captured image into frames.
3. Image pre-processing using Histogram Equalization.
4. Edge detection of the hand by using an algorithm like Canny Edge Detection.
5. Enlargement of the edges of regions of foreground pixels by using Dilation to get a continuous edge.
6. Filling of the object enclosed by the edge.
7. Storing the boundary of the object in a linear array.
8. Vectorization operation performed for every pixel on the boundary.
9. Detection of the fingertips.
10. Tracking of the fingertips in consecutive frames to determine the motion.

11. Identification of the gesture based the motion.
12. Insertion of the input stream into the normal input path of the computing device.

Fig. 1 shows the block diagram using the Edge Detection Approach. The images are first captured using a webcam, separated into frames and converted into grayscale format. The contrast is then improved using Histogram Equalization. After the edges are detected, the images are dilated to fill up the broken edges. The images are then filled up using `bwboundaries` function in MATLAB, and the boundary pixels are detected and stored sequentially in a linear array. The fingertips are then detected using vectorization technique and the gesture is recognized by the system depending on the relative movement of the fingertips in the different frames.

B. SKIN DETECTION

Following steps are used for skin detection

1. Image capturing using a webcam or the front camera of the mobile phone.
2. Converting the captured image into frames.
3. Skin color detection using hue and saturation values of various possible skin tones.
4. Storing the boundary of the object in a linear array.
5. Vectorization operation performed for every pixel on the boundary.
6. Detection of the fingertips.
7. Tracking of the fingertips in consecutive frames to determine the motion.
8. Identification of the gesture based the motion.
9. Insertion of the input stream into the normal input path of the computing device.

Figure 2 shows the block diagram using Skin Detection Algorithm. In this approach, after the images are separated into frames, the skin detection algorithm is applied. The closed contour of the fingers is identified in this technique even in the presence of a noisy background and hence eliminates the need of a white background which was required in the first approach. Then the vectorization technique is used to detect the fingertips and the gesture is recognized by the system depending on the relative movement of the fingertips in the different frames.

IV. DESCRIPTION OF ALGORITHMS

Working of hand gesture recognition process using MATLAB is discussed as follows;

A. VIDEO CAPTURING USING A CAMERA

The image will be captured with the help of a single web camera, which will then lead to the image pre-processing stage. In case of mobile phones, it will be captured by the front camera of the mobile phones.

B. FRAME SEPARATION

The frames of the captured video are saved as images. A MATLAB function is used for this purpose.

C. OBJECT TRACKING

We use a local image co-ordinate scheme for determining the fingertips. The co-ordinate system is determined by the position of the fingertip at the start of the gesture. In simple words, the co-ordinate system is established in the first frame of a sequence of gestures, and then is kept constant for the subsequent frames. Thus, the need of having a common co-ordinate system for all images is eliminated. This sort of a system emulates a human eye i.e. the brain perceives any gesture irrespective of the background.

D. IMAGE PREPROCESSING

This block will basically concentrate on Histogram Equalization. In this stage we aim to increase the contrast among neighboring pixels, as shown in Fig. 3(b). The lowest colored pixel value is reduced to zero and the greatest colored pixel value is made to value 255. The other neighboring pixel values are averaged and spaced out in a similar manner. This helps us to locate our object of interest from the background. Figure 3 shows the Histogram Equalization Process

E. EDGE DETECTION

After converting this image into grayscale image edge detection is applied. Here we find the points of the image where there are sharp edges or discontinuities or where sharp change in brightness is encountered. We will apply the Canny Edge Detection Algorithm for the purpose of detecting points at which image brightness changes sharply or formally, there are more discontinuities. Figure 4 shows the edge-detected hand.

F. IMAGE FILLING AND BOUNDARY DETECTION

From the hand contour obtained from the preprocessing steps, the feature of interest is the set of fingertips, which, in turn, is a subset of the boundary of the hand. We use *bwboundaries*, a MATLAB function to store the boundary of the hand contour in a linear array, formed sequentially from the topmost and leftmost boundary pixel, which is on. *bwboundaries* detects boundaries of filled images or holes within filled objects. Thus, we fill the continuous edge of the hand contour with white pixels as shown in Figure.5.

Further, we detect boundaries of all objects in a cell array, each cell corresponding to the boundary of one object, and each element in every cell corresponding to a pixel on the boundary of that object. Since the hand should ideally correspond to the largest object in the image, we detect the largest cell array for use in vectorization. This eliminates any adverse effects noisy background might have on fingertip detection.

G. VECTORIZATION

In order to reduce computing complexity we define the angle $C(i)$ between two vectors $[P(i-k), p(i)]$ and $[P(i), p(i+k)]$ as curvature, where k is a constant. The points along the edge where the curvature reached a local extreme, that is the local features, are then identified. Some of these local features are labeled as "peak" or "valley".

We use this algorithm to compute curvatures at every point, and thus detect positions of the fingertip in the boundary detected hand contour as shown in Figure 6 It shows Yellow vectors denote curvatures belonging to a fingertip; green vectors denote curvatures, which do not belong to a fingertip.

H. FINGERTIP TRACKING AND GESTURE IDENTIFICATION

In fingertip tracking and Gesture Identification; we can use ANN algorithm, to train the system and accordingly give us the necessary output.

I. Artificial Neural Network

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this *supervised learning* to train a network.

Artificial neurons are much simpler than the biological neuron; Figure 7 shows the basics of artificial neurons.

The ability of neural networks to discover nonlinear relationships in input data makes them ideal for modeling nonlinear dynamic systems such as the stock market. [16]

2. Training a Neural Network

A neural network must be trained on some input data. The two major problems in implementing this training discussed are:

1. Defining the set of input to be used (the learning environment)
2. Deciding on an algorithm to train the network

3. The Learning Environment

Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings. There are two modes of learning:

1. Supervised learning.
2. Unsupervised learning

3.1 Supervised learning

Supervised learning or Associative learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by an external teacher, or by the system which contains the neural network (self-supervised).

Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). The supervised learning paradigm is also applicable to sequential data (e.g., for speech and gesture recognition). This can be thought of as learning with a "teacher," in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

3.2 Unsupervised learning

Unsupervised learning or Self-organization in which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

The supervised training methods are commonly used, but other networks can be obtained from nsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. [17]

4. Network Training

Training a network involves presenting input patterns in a way so that the system minimizes its error and improves its performance. The training algorithm may vary depending on the network architecture, but the most common training algorithm used when designing financial neural networks is the *backpropagation*

algorithm. Figure 8 shows the Backpropagation Network.

Backpropagation is the process of backpropagating errors through the system from the output layer towards the input layer during training. The most common network architecture for financial neural networks is a multilayer Feedforward network trained using backpropagation.

The feedforward neural network was the first and arguably simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

Fig 9.shows a representation of a simple feed-forward Neural Network with four inputs, one hidden layer and four outputs. Neural networks learn by changing their weights.

During feed forward, each input unit receives an input signal and broadcasts this signal to each of the hidden units. Each hidden unit then computes its activation and sends its signal to each output unit. Each output unit computes its activation to form the response for the given input pattern.

During back propagation of associated error, the output from the output units are compared with the target value associated with output unit and error is calculated. This error is then backing propagated back to the hidden units and similarly the errors from hidden layers are back propagated to the input layer.

V. CONCLUSION

Hand gesture recognition based man-machine interface is being developed vigorously in recent years. Gesture recognition is also important for developing alternative human-computer interaction modalities. It enables human to interface with machine in a more natural way.

MATLAB provides the better solution for hand gesture recognition. The Canny Edge Detection Algorithm for the purpose of detecting points at which image brightness changes sharply or formally. We used ANN algorithm for gesture identification. ANN has a fast computational ability. Edge detection and ANN provides good and powerful solution for hand gesture recognition using MATLAB.

REFERENCES

- [1]. Sebastian Marcel, Oliver Bernier, Jean Emmanuel Viallet and Daniel Collobert, " Hand Gesture Recognition using Input – Output Hidden Markov Models", Proc. of the Fourth IEEE International

Conference on Automatic Face and Gesture Recognition, pp. 456 - 461 2000.

[2]. Xia Liu and Kikuo Fujimura, “ Hand Gesture Recognition using Depth Data”, Proc. of the Sixth IEEE International conference on automatic Face and Gesture Recognition, pp. 529-534, 2004.

[3]. L. Bretzner, I. Laptev, and T. Lindeberg, “Hand Gesture using multi-scale color features, hierarchical models and particle filtering”, Proc. of the Fifth International conference on Automatic Face and Gesture Recognition, pp. 423- 428, 2003.

[4]. Pavlovic, et al. visual interpretation of hand gesture for human-computer interaction: a review, IEEE Trans. On Pattern anal. Mach. Intel. 19(7), pp 677-695, 1997.

[5]. Attila Licsar and Tamas Sziranyi, “Supervised training based hand gesture recognition system”, Proc. of the 16th International Conference on Pattern Recognition, Vol. 3, pp 30999 – 31003, 2002.

[6]. E.Stergiopoulou and N.Papamarkos: “A New Technique on Hand Gesture Recognition”, Proc of the IEEE International Conference on Image Processing, 2657-2660, 2006.

[7]. Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang and Toshiaki Ejima : “Hand Gesture Recognition Using Hidden Markov Models”, Proc. of the IEEE International conference on Systems, Man and Cybernetics, vol 5, pp. 4232 -4235, 1997.

[8]. Meide Zhao, Francis K.H. Quek, Member, IEEE, and Xindong Wu, Senior Member, IEEE : “RIEVL: Recursive Induction Learning in Hand Gesture Recognition”, IEEE Transactions on Pattern Analysis and machine intelligence, vol. 20, no. 11, November 1998.

[9]. Yikai Fang, Jian Cheng, Kongqiao Wang and Hanqing Lu, “Hand Gesture Recognition Using Fast Multi-scale Analysis”, Proc. of the Fourth International Conference on Image and Graphics, pp 694-698, 2007.

[10]. Chris Joslin, Ayman El-Sawah, Qing chen, Nicolas Georganas, “ Dynamic Gesture Recognition”, Proc. of the Instrumental and Measurement Technology Conference, pp 1706-1710, 2005.

[11]. Timi Ojala, Matti Pietikainen and Topi Maenpaa, “Multi-resolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971-987, 2002.

[12]. Prateem Chakraborty, Prashant Sarawgi, Ankit Mehrotra, Gaurav Agarwal, Ratika Pradhan, “Hand

Gesture Recognition: A Comparative Study”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong.

[13]. Daniel Thalmann, “Gesture Recognition Motion Capture, Motion Retargeting, and Action Recognition”, EPFL – VRlab.

[14]. Aditya Ramamoorthy et al. “Recognition of dynamic hand gestures “, page 1-13. Department of Electrical Engineering IIT New Delhi-110016 India. October 2002.

[15]. http://en.wikipedia.org/wiki/Gesture_recognition

[16]. Tin Hninn Hninn Maung ,”Real-Time Hand Tracking and Gesture Recognition System Using Neural Networks”, World Academy of Science, Engineering and Technology 50 2009

[17]. http://www.aihorizon.com/essays/generalai/supervised_unsupervised_machine_learning.htm

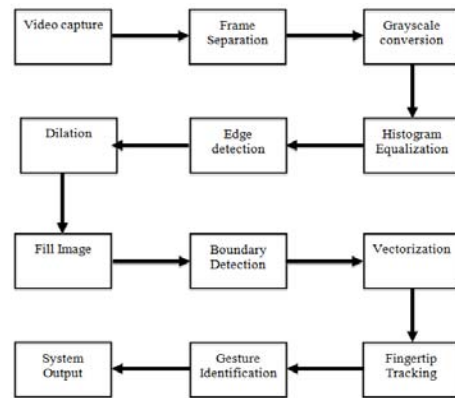


Figure1. Block Diagram Using Edge Detection Algorithm

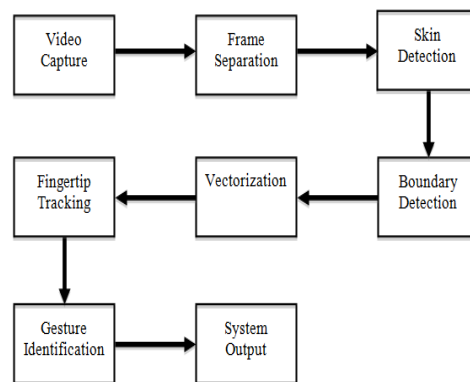


Figure 2. Block Diagram Using Skin Detection Algorithm

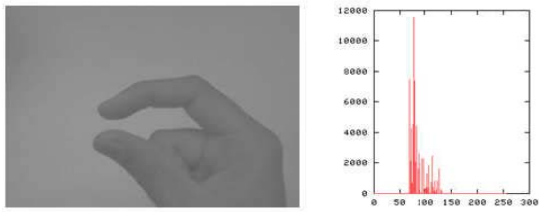


Fig (a) Before Histogram Equalization



Figure 3. Histogram Equalization (a) Before and (b) After Histogram Equalization

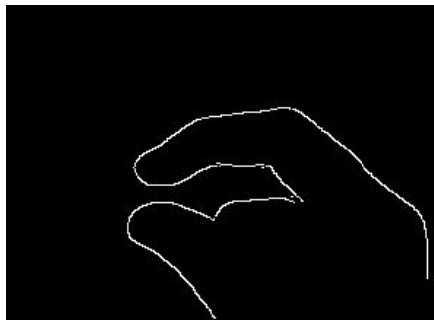


Figure 4. After Edge Detection



Figure 5. Filled Image

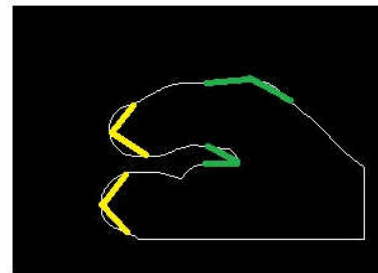


Figure 6. Vectorization: Yellow vectors denote curvatures belonging to a fingertip; green vectors denote curvatures, which do not belong to a fingertip.

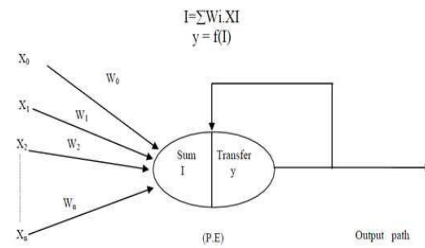


Figure 7. Artificial Neuron

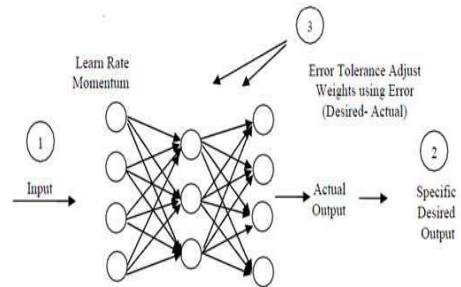


Figure 8. Backpropagation Network

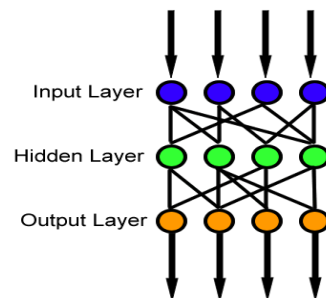


Figure 9. A simple feed-forward Neural Net