Different Hand Gesture Recognition Techniques Using Perceptron Network

Nidhi Chauhan
Department of Computer Science & Engg.
Suresh Gyan Vihar University,
Jaipur(Raj.) Email:
nidhi99.chauhan@gmail.com

Abstract

Previous systems have used datagloves, markers or any other input devices for interaction. Hand gesture recognition system provides an alternative to interface devices for human computer interaction (HCI). There are several hand gesture recognition techniques are proposed. In this paper various gesture recognition algorithms are used with Perceptron network.

It will recognize static and dynamic hand gestures. In this paper, overview of various different methods for hand gesture are discussed. It has overview of the recognition processes that is edge detection algorithm and skin color detection algorithm. In this, trained and untrained gestures are used for training and testing respectively and the results yields a satisfactory recognition rate.

Keywords - Perceptron neural network, Hand gesture recognition, Edge detection algorithm.

1) INTRODUCTION

Hand gesture recognition using visual devices has a number of applications in virtual reality (VR), human computer interaction (HCI) and machine control. Vision based interfaces are feasible and at the present user are more friendlier and more interactive with the machines without using devices for input. Computer recognition of hand gestures provide natural computer interface and reduces response time and it should provide no appreciable delay between when he or she makes a gesture motion and computer responds.

Hand gesture research can be classified into three categories. First category is glove based analysis that employs sensors attached to a glove that transduces finger flexions into electrical signals for determining the hand posture. Second category is vision based analysis that is based on human beings perceive information about their surroundings. The third category is analysis of drawing gestures that make use of
stylus as an input device. Visual sensing is most often used for gesture recognition than mechanical sensing because it is more practical and improve reliability, accuracy and electromagnetic noise.

Many methods for hand gesture recognition using visual analysis have been proposed for hand gesture recognition. Sebastiean 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 approached uses 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 Yangc 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].

2) HAND GESTURE RECOGNITION ALGORITHMS

Hand gesture is the meaningful movements or expressions of the hand with the intent to convey information or interact with environment. Hand gesture recognition is the process through which we can communicate with the machines, so the hand gesture recognition techniques enables human-computer interaction. Gesture recognition is static, dynamic and both. In this paper we are discussing static and dynamic system for hand gesture.

Gesture recognition is also important for developing alternative human computer interaction modalities that enables human to interface with machine in more convenient way.

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. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. It includes among others image processing and neural networks toolboxes.

Typical uses include:
_ Math and computation
_ Algorithm development
_ Modeling, simulation, and prototyping
_ Data analysis, exploration, and visualization
_ Scientific and engineering graphics
_ Application development, including Graphical User Interface building

We use two algorithms for hand gesture recognition are edge detection and skin detection algorithm. In this paper we are using edge detection algorithm for static and dynamic hand gesture recognition. The steps that are involved in edge detection are:

1. Capturing of image using webcam.
2. Captured image is then separated into frames.
3. The image is then preprocessed and converted into grayscale format.
4. Detection of edge of hand by using appropriate Edge Detection algorithm.
5. Images are dilated to fill up the broken edges and to get continuous edge.
6. Filling of images enclosed by the edge.
7. Boundary pixels are detected and stored sequentially in a linear array.

8. Vectorization performed for every boundary pixel.
9. Fingertips are detected.
10. Determine the motion by tracking fingertips.
11. Identification of the gesture.
12. Insertion of the input stream into the normal input path of the computing device.

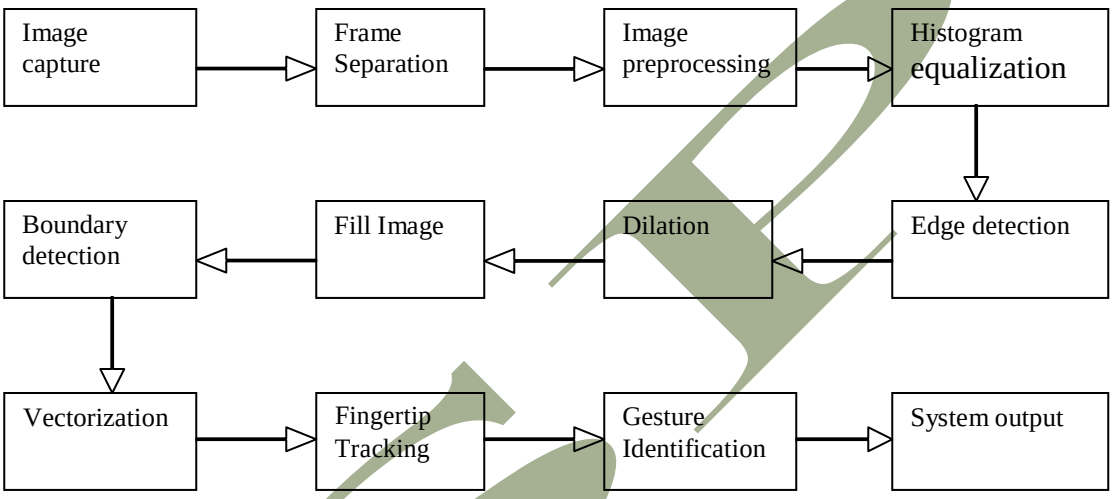

Fig1:- Block diagram of Edge Detection Algorithm.

Description of each step of Edge detection algorithm:

1) Video Capturing

The image will be captured by using webcam, which will use for further processing. If we are using mobile phone, then the image will be captured by using front camera of mobile.

2) Frame Separation

The captured video are separated into frames and stored in the frame pattern.

3) Object Tracking And Image Preprocessing

Object Tracking is done by using local co-ordinate system. The co-ordinate system is used for find out the position of the fingertip at the start of the hand gesture. This system is established in the first frame of sequences of gesture and then kept constant for the subsequent frames. This system emulates a human eye.

After object tracking image is preprocessed by using Histogram Equalization. In this step we focus on interested object from the background by increasing the contrast among the neighboring pixels and neighboring pixels values are averaged. The lowest colored pixel value is reduced to zero and greatest colored pixel value is made to 255.

Fig2. Before Histogram Equalization

Fig3. After Histogram Equalization

4) 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.

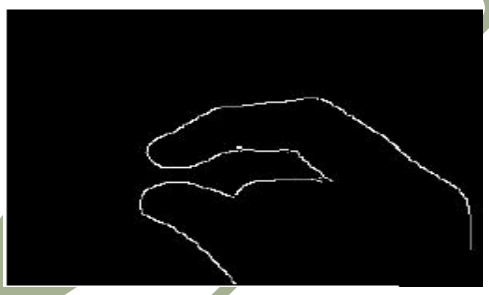

Fig4. After Edge Detection

5) 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.

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.

6) Vectorization

In order to reduce computing complexity we define the angle *C(i)* between two vectors [$P(i\text{-}k)$, $p(i)$] 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 5 It shows Yellow vectors denote curvatures belonging to a fingertip; green vectors denote curvatures, which do not belong to a fingertip.

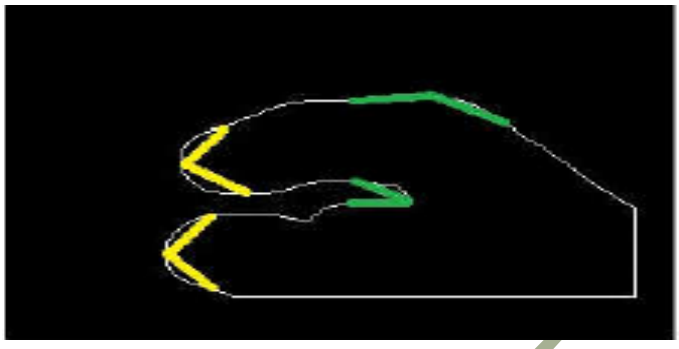

Fig5. Vectorization: Yellow vectors denote curvatures belonging to a Fingertip; green vectors denote curvatures which do not belong to a fingertip.

7) Fingertip Tracking and Gesture Identification

We can use perceptron network for fingertip tracking and gesture identification , that is used to train the system and gives the required output.

A) Perceptron 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 learnin*g to train a network. The supervised learning is commonly used, but other networks can be obtained from *unsupervised learning* or direct design methods.

Neural network have been trained to perform complex functions in various fields of application including pattern recognition, classification, speech, vision and control system.

Perceptron is a program that learns concepts, i.e. it can learn a respond with true or false for inputs. The perceptron (artificial neuron) consists of numerical value multiplied by a weight plus bias , the perceptron fires the output only when the total signal of the input exceeds a specific threshold value. The activation function controls the magnitude of the output, and then the output is fed to other perceptron in the network.

B) The Learning Modes

Neural network will be trained to solve problems that are difficult for computers or human being. There are two modes of learning:

i) Supervised Learning
ii) Unsupervised Learning

i) Supervised Learning:

Supervised learning or associative learning in which the network is trained by providing it with input and output pairs, that is given by system or external teacher. In this system, its predictions are compared with the target output and learns from its mistakes. Inputs are applied to the input layer, then it passes to the next nodes and weights are summed up or weakened. In Supervised learning, the predicted output is compared to the actual output. If the predicted output is equal to actual output, no change is made to weights. But if predicted output is higher or lower than the actual output the error is propagated back to the system. The Supervised learning is applied to sequential data.[12]

ii) 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 supervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. [12]

C) 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***.

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.

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.

3) CONCLUSION

Hand gesture recognition system provides the natural and convenient way to communicate with system or computer. There is no need of any special hardware for giving input. So, hand gesture recognition system provides human computer interaction modalities.

In this paper we used different algorithms for development of hand gesture recognition system. We used edge detection algorithm for detection of gestures. Perceptron network is used for gesture identification and it helps to solve complex computational problem. Edge Detection algorithm and perceptron network together give better and powerful solution. MATLAB is used for hand gesture recognition system, which improve performance and provides better solution.

4) 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 Yangc 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. http://www.aihorizon.com/essays/generalai/supervised_unsupervised_machine_learning.htm.