Real-Time Robotic Arm Control using Static Hand Gestures: An Approach towards the Natural Human Machine Interaction

Bhavin Changela
Head of Department
Department of Biomedical Engineering
A.V. Parekh Technical Institute, Rajkot

Atulkumar Kataria
Lecturer
Department of Biomedical Engineering
A.V. Parekh Technical Institute, Rajkot

Abstract

The main objective of this paper is to explore the utility of a human hand gesture for natural Human Machine Interaction. Hand gestures are a powerful communication channel and can be used as an interface device to achieve a natural and immersive Human Machine Interaction. In this paper, we have proposed a vision based system for controlling a robotic arm, in which system is able to recognize a set of fourteen specific static hand gestures and use them to control a 5 degree of freedom robotic arm. In the proposed method hand gesture images are acquired using HD web-camera and then passed through three stages: Image preprocessing, feature extraction, and recognition. For gesture recognition multi-layer feed-forward neural network classifier with back-propagation learning algorithm is used. Once a hand gesture is recognized, an appropriate command is sent to a robotic arm to perform a pre-defined task. The proposed system has been extensively tested with success. The proposed recognition technique for static hand gestures has been implemented in MATLAB. The average performance of the system to recognize hand gestures is more than 98% and the robotic arm is able to do jobs by using hand gesture commands as its input.

Keywords- Static Hand Gesture, Robotic Arm, Neural Network, Human Machine Interaction, Computer Vision

I. INTRODUCTION

Hand gestures are expressive meaningful movements of a hand with the purpose of information transfer or interaction with the environment. Hand gestures provide a powerful communication channel and people all over the world use gestures for information transfer in their everyday life. Human hand gestures also can be used as an interface device to achieve natural Human Machine Interaction (HMI). Nowadays many researchers are working on different application of human hand gestures to make interactions more easy, natural and convenient without wearing any extra device [1]. Variety of applications of hand gesture including sign language recognition, robot control, smart home automation, wheel chair control, table top interaction, interaction with Humanoid Robot and game playing, have been presented by various researchers [2-7]. In gesture based Human-Machine Interaction the main objective is to identify a particular human gesture and on the basis of that an appropriate command is sent to the machine for execution of a particular action. However to achieve this goal, it’s required to implement a successfully hand gesture recognition technique.

Hand gestures are generally classified into two class, static hand gestures and dynamic hand gestures. Static hand gestures are described in term of hand pose, which is also known as hand posture. Whereas the dynamic hand gestures are described in terms of movements of the hand [8]. There are mainly two approaches for hand gesture recognition, Hand glove-based approach and computer vision-based approach. In hand glove-based technique, number of sensors is mounted on the glove, which provides the information for determining the hand position [9]. In vision-based techniques, hand gestures are recognized with the help of computer and video camera, by extracting, processing, and analyzing the relevant information from a hand image or sequence of hand images. Glove-based technique provides more accurate result in hand gesture recognition compare to vision-based technique, but it is more expensive and unnatural, whereas vision based gesture recognition enables humans to communicate with the machine and interact naturally without any mechanical devices such as keyboard, mouse, joystick or game-pad. Also user can interact with machine without the technical knowledge of any hardware and software [15].

In this work, we have proposed a vision based system which can recognize a specific static hand gestures and use them to control a 5 degree of freedom robotic arm. The overall system consists of two main tasks: Static hand gesture recognition and robotic arm control.

II. STATIC HAND GESTURE RECOGNITION

The main purpose of the proposed hand gesture recognition method is to recognize a set of 14 selected static hand gestures using soft computing approach. The sample images of selected hand gestures are shown in figure 1. In the proposed technique, a multi-
layer feed-forward neural network classifier with back-propagation learning algorithm is used for hand gesture recognition. The overall technique to acquire hand gestures and recognize them using Neural Network is divided into four subparts:
1) Image Acquisition
2) Image preprocessing
3) Feature Extraction and
4) Neural Network Training or Testing,
As illustrate in figure 2.

During the training phase extracted features are used to train the neural network and during the recognition phase extracted features are given to the network as an input. Neural network is train using the supervised back-propagation learning algorithm.

**A. Image Acquisition**
Real-time hand gesture images are acquired using Logitech HD 310 web-camera shown in fig 3. This web-camera provides image capture with maximum resolution of 1024x576, in our approach the color images of hand gesture with resolution of 320x240 is acquired with this camera. In the proposed technique it is assumed that gestures are made with right hand, the input images include exactly one hand, the arm is roughly vertical and facing palm towards the camera and the image background is white with proper lighting. Top view of image acquisition setup can be visualized in figure 3.

**B. Image Preprocessing**
After acquiring the color image of hand gesture with resolution of 320x240, Image preprocessing steps are applied to extract the hand region from its background and prepare the image for the feature extraction stage. In the proposed method image
Real-Time Robotic Arm Control using Static Hand Gestures: An Approach towards the Natural Human Machine Interaction

preprocessing steps include, Hand region detection, Image Binarization, Image Filtering, Region of Interest Extraction (ROI), Image resizing and finally Hand Contour Detection.

Hand region detection is achieved using color segmentation, in which image pixels are classify into skin color and non-skin color clusters. First of all the RGB color image is converted into YCrCb color space, using the following equations, which gives information of luminance and chrominance from RGB color values. Luminance represents the gray-scale information, while chrominance represents the color information.

\[
Y = 16 + 0.299 R + 0.587 G + 0.114 B \quad (1)
\]
\[
Cb = 128 - 0.1687 R - 0.3312 G + 0.5 B \quad (2)
\]
\[
Cr = 128 + 0.5 R + 0.4186 G + 0.0813 B \quad (3)
\]

After this the classification of the pixels of the input image into skin color and non-skin color clusters is accomplished by using Otsu's thresholding technique that use the information of a skin color distribution map in the YCrCb color space. The Otsu's algorithm assumes that the image to be threshold contains two classes of pixels (e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread is minimal [10]. Following are the steps involve in Otsu’s thresholding algorithm.

1) Compute histogram and probabilities of each intensity level
2) Set up initial \( \omega_i(0) \) and \( \mu_i(0) \), weight and mean
3) Step through all possible thresholds \( t = 1 \ldots \text{maximum intensity} \), Update \( \omega_i \) and \( \mu_i \) and Compute \( \sigma_b^2(t) \), between-class variance
4) Desired threshold corresponds to the maximum \( \sigma_b^2(t) \)
5) You can compute two maxima and two corresponding thresholds. \( \sigma_b^1(t) \) is the greater max and \( \sigma_b^2(t) \) is the greater or equal maximum
6) Desired threshold \( = \text{threshold}_1 + \text{threshold}_2 / 2 \)

By using calculated threshold value image is converted into binary form, by turning all pixels below threshold to zero and all pixels above that threshold to one. Final output of this skin color segmentation process is a binary image, in which the hand region, that is the region of interest, turns white and the background black as shown in figure 4.

![Fig. 4: Image preprocessing steps](image)

It is clearly observe from the extracted binary image that the image had some dark pixels inside the hand region, some white pixels outside the hand region and rough edges. So the next step is to apply median filtering to remove the white spots and make edges smoother. Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. A median filter is more effective than other filters when the goal is to simultaneously reduce noise and preserve edges [11, 12]. Next to this, region feeling operation is applied to the segmented binary image to remove black region inside the white foreground.

Once the noiseless binary image has been accomplished, the region of interest (ROI) is extracted, in which the extra part other than hand gesture region is cropped out to make image more efficient. After that the hand gesture image is resized and converted into the image with resolution 50x50, such that the aspect ratio of original image does not change. Image resizing makes the system faster, this operation will reduce the negative effects of the size change. Finally the contour of hand gesture region is extracted by applying morphological operation and make image ready for feature extraction. All the image preprocessing steps are illustrated in figure 5.

C. Feature Extraction

The objective of the feature extraction stage is to discriminate the most relevant characteristics of the hand gesture from the image. The selection of good features can strongly affect the classification performance and reduce the computational time. In the proposed method two types of features are extracted, namely, the chain code of hand contour as a geometric feature and Sum of the pixel values of each row and column as assistant general features.
The chain code was first proposed by H. Freeman and therefore sometimes referred to as Freeman code or Freeman chain code [13]. A chain code simply represents the movements of boundary segment in terms of an integer number, as shown in fig 5. A boundary code formed as a sequence of such directional numbers is referred to as a Freeman chain code. Finally the feature vector consisting of 350 elements is extracted. In feature vector first 250 elements represents the chain code of hand contour and next 100 elements represents the sum of pixel value of each row and column.[14]

![Chain code representation](image)

**D. Neural Network based Classifier**

The right choice of the classification algorithm to be used in a gesture recognition system is highly dependent on the properties and the format of the features that represent the gesture image. In this work a standard multi-layer feed-forward neural network is used to classify the gestures. The network consists of four layers; the first layer consists of neurons that are responsible for inputting a hand gesture features into the neural network. The second and third layers are hidden layers, this layer allows neural network to perform the error reduction necessary to successfully achieve the desired output. The final layer is the output layer with fourteen nodes.

![Neural Network architecture](image)

Typically the number of neurons in output layer is determined by the size of the set of desired outputs, with each possible output being represented by a separate neuron. The structure of this particular neural network is illustrated in Fig 6. The basic operation of an artificial neuron involves summing its weighted input signal and applies an activation function to get output. We have used a bipolar-sigmoid function which holds the output always between -1 and 1. The steps involve in the back-propagation algorithm repeatedly iterates over all training examples. Each iteration is called an epoch. For each training example, it executes two parts of code. The first part of the code propagates the inputs forward through the neural network to calculate the outputs of all units. The second part of the code propagates the error backward through the neural network and updates the weights of all units using gradient descent with a user-provided learning rate.

The hand gesture recognition process consists of two phases: training (learning) phase and classification (testing) phase, as shown in figure 3. During the training phase, neural network works on the gesture images used to train the neural network; whereas during the testing phase, neural network works on the gesture images used to test the classification ability of network.
III. MATHEMATICS

A. Back Propagation learning Algorithm

Input: $x_k[e]$, Desired output $t_i[e]$, Learning rate $\alpha$;

Network: two-hidden-layer feed forward neural network with weights $w_{kj}$, $w_{ji}$ and $w_{li}$;

- Propagate the input forward
  1) for each hidden unit of layer 1 $u_j$ do:
     \[ a_j = \sum_k w_{kj} x_k[e] \text{ and } o_j = \frac{2}{1 + e^{-2\alpha_j}} - 1 \]

  2) for each hidden unit of layer 2 $u_l$ do:
     \[ a_l = \sum_j w_{jl} x_j[e] \text{ and } o_l = \frac{2}{1 + e^{-2\alpha_l}} - 1 \]

  3) for each output unit $u_i$ do:
     \[ a_i = \sum_l w_{li} x_l[e] \text{ and } o_i = \frac{2}{1 + e^{-2\alpha_i}} - 1 \]

- Propagate the error backward
  4) for each output unit $u_i$ do: $\delta_i = o_i (1-o_i)(t_i[e]-o_i)$

  5) for each hidden unit of layer 2 $u_l$ do: $w_{li} = w_{li} + \alpha \delta_i o_l$

  6) for each hidden unit of layer 2 $u_l$ do: $\delta_l = o_l (1-o_l)(\sum_i w_{li} \delta_i)$

  7) for each hidden unit of layer 1 $u_j$ do: $w_{jl} = w_{jl} + \alpha \delta_i o_j$

  8) for each hidden unit of layer 1 $u_j$ do: $\delta_j = o_j (1-o_j)(\sum_l w_{jl} \delta_l)$

  9) for each input unit $u_k$ do: $w_{kj} = w_{kj} + \alpha \delta_i o_k[e]$  

All the steps are repeatedly iterates over all training examples until performance goal is achieved.

IV. RESULT

In the proposed method for the creation of static hand gesture database, instead of taking gesture photographs, gesture videos are taken and static gesture frames are obtained by using video player. In gesture video, gestures are varied in both position and orientation. Total 140 gesture images of my own hand, include 10 images of each gesture, are used to train the neural network and 560 gesture images of other user, include 2 images of each gesture are used for testing purpose. The proposed recognition techniques for static hand gestures have been implemented in MATLAB. The summary of all recognition results and the recognition rates for each of the fourteen static hand gestures is presented in Table 1. Gesture recognition rate is varying from 84% to 100% and average recognition rate is 95.04%. System is also tested in real time and successfully controlled a robotic arm using hand gesture and the system is opened and operated when it’s recognized registered user.

<table>
<thead>
<tr>
<th>Gesture Number</th>
<th>Number of Test Gestures</th>
<th>Successful Recognition</th>
<th>Recognition rate %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>39</td>
<td>97.5</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>8</td>
<td>40</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>39</td>
<td>97.5</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>14</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>560</td>
<td>550</td>
<td>98.21</td>
</tr>
</tbody>
</table>
V. ROBOTIC ARM CONTROL

For implementation of robotic arm control most commonly used educational robot i.e. Intellitech SCORBOT has been chosen and we have used the Matlab toolbox developed by Carl Wick, Joel Esposito and Ken Knowles, for robotic arm control [14]. The tool box uses two types of coordinates: Cartesian and Joint, as illustrated in Figure 5.3. Cartesian coordinates are specified as a 5 X 1 vector of the form [X Y Z Pitch Roll]. Commands that utilize Cartesian coordinates contain the characters ”Cart” in the function name. (ex. ScorCartMove( [X Y Z P R] ). Joint coordinates are specified as a 5 X 1 vectors of angles the form [Base Shoulder Elbow Pitch Roll]. Commands that utilize Joint coordinates contain the characters ”Jt” in the function name (ex. ScorJtMove( [B S E P R] ). An example of a very basic MATLAB program which uses the MTIS wrapper functions to pick up an object and determine its size is included below [16-17].

ScorInit; % Loads the DLL and initializes USB
ScorHome; % The Scorbot must be homed before beginning
ScorSetSpeed(80); % Set speed at 80 percent of max
ScorCartMove( [40 0 30 -90 0 ]); % Moves to a position 40 cms in front of robot with end effector pointing down
ScorSetGripper(-1); % Opens gripper fully
ScorDeltaCartMove( [0 0 -10 0 0 ]); % Moves end effector down 10 cms to pick up object

The overall robotic system is comprised of three hardware components: Controller-USB, SCORBOT robot arm and Computer. The computer is connected to Controller-USB via a USB cable. The robot is connected to Controller-USB by a proprietary interface cable.

Total 14 different hand gestures are related to 10 different movements of robotic arm and other operation. Whenever a particular gesture is recognized by Neural Network, the instruction set corresponding to that gesture is identified and passed to robot for execution. In this way the robotic system can be controlled by hand gesture using live camera. Fig 8 shows the hand gesture commands used to control the robotic arm as per the requirement.

![Fig. 7: SCORBOT Cartesian coordinate and Joint coordinate](image-url)

![Fig. 8: Hand Gesture commands for Robotic Arm control](image-url)
VI. SYSTEM IMPLEMENTATION

We have implemented all the Image processing algorithm and functions for robotic arm control in MATLAB environment. Whole system is implemented in robotics laboratory of our university as shown in fig 9. We have also design and develop an interactive MATLAB User Interface for easy system operation. Fig 10 illustrates the GUI design for Hand gesture based robotic arm control. Here we have use secondary camera just to visualize the movement of the robotic arm, so that user can observe the movement of robotic arm corresponding to the given hand gesture. Following is the procedure for Hand Gesture based Robotic Arm control:
1) Press Initialization button for Hardware initialization
2) Press Homing button to perform robotic arm homing
3) Put hand in initial position and then press Start button
4) Perform task using hand gesture commands
5) Press stop button after completion of task

Fig. 9: Laboratory setup

Fig. 10: Matlab GUI for Hand Gesture based Robotic Arm control

VII. CONCLUSION

This paper presents a computer vision algorithm for static hand gesture recognition in real-time, with one camera as the input device. The importance of hand gesture recognition lies in building efficient human-machine interaction. Its applications range from sign language recognition through smart home automation system to virtual reality. In this work, to demonstrate the effectiveness of human hand gesture for human-Robot interaction, a gesture-based robotic arm control system is implemented. As artificial neural networks have proved to be an extremely useful approach to pattern classification, a supervised multi-layer feed-forward neural network is used for gesture recognition, with hand contour based features. The proposed system has been extensively tested with success. The average performance of the system to recognize hand gestures is more than 98% and the robotic arm is able to do jobs by using hand gesture commands as its input. In future faster recognition of hand gesture and controlling of robotic arm can be implemented using more advanced image processing algorithms.
ACKNOWLEDGEMENT

The authors appreciate the constant effort from the students of the department for supporting in collection of the hand gesture database. We also acknowledge the constant support by the faculties of the Biomedical Department for the implementation of the system using MATLAB platform.

REFERENCES