Hand Gesture Recognition System

A Thesis submitted in partial fulfillment of the Requirements for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

Lagnajeet Sahoo

Roll no. 111EC0186

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

May 2015

Hand Gesture Recognition System

Thesis Submitted in
May 2015
To the department of
Electronics and Communication Engineering
Of
National Institute of Technology, Rourkela
is a partial fulfillment of the requirement
for the degree of
Bachelor of Technology
by

Lagnajeet Sahoo
(111EC0186)
Under the guidance of
Prof. Samit Ari

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

May 2015

Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha India. www.nitrkl.ac.in

DECLARATION

I, Lagnajeet Sahoo (Roll No. 111EC0186) understand that plagiarism is defined as any one or the combination of the following

- Uncited verbatim copying of individual sentences, paragraphs or illustrations (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
- Uncited improper paraphrasing of pages or paragraphs (changing a few words or phrases, or rearranging the original sentence order)
- Cited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did or wrote what.

I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly cited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the thesis may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Date:

Lagnajeet Sahoo
Roll no. 111EC0186
B. Tech in Department of Electronics and Communication
NIT Rourkela

Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha India. www.nitrkl.ac.in

CERTIFICATE

This is to certify that the work in the thesis entitled Hand Gesture Recognition System by Lagnajeet Sahoo with roll no. 111EC0186 is a record of an original research work carried out by him during 2014 - 2015 under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering, National Institute of Technology, Rourkela. Neither this thesis nor any part of it, to the best of my knowledge, has been submitted for any degree or diploma elsewhere.

Place: Rourkela — Dr. Samit Ari

Date: — Asst. Professor

Lagnajeet Sahoo
111EC0186

ACKNOWLEDGEMENT

"The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires."

— WILLIAM ARTHUR WARD

It would be an honor for me to acknowledge that without the endless support and valuable advice of my guide, Prof. Samit Ari, this work would never had been successfully completed. I have utmost respect for him. Because of his guidance, it was possible for me to surpass all the technical difficulties. His advice were very valuable in times when I was stuck or had hit a dead end.

I would also take this opportunity to acknowledge the support of Prof. Kamalakanta Mahapatra, Prof. A.K. Swain, Prof. S.M. Hiremath and Prof. U.K. Sahoo for scrutinizing my work during the period of evaluation, which only compelled me to produce more finer work and put more labor in my project.

I am thankful to all my friends who have helped in every possible way to keep me going on in my academic life.

Last but not least I would convey my heartfelt gratitude towards my parents for unconditional support throughout my entire life and without them I would not be here.

Lagnajeet Sahoo

Contents

LIST OF FIGURES

Chapter 1

Chapter 2

LIST OF TABLES

Chapter 1

Chapter 2

LIST OF EQUATIONS

Chapter 1

Chapter 2

ABSTRACT

Hand Gesture Recognition is a well-researched topic in the community of Machine-Learning, Computer Graphics and Image Processing. The system which are based on Recognition technology follow mathematically rich and complicated algorithms whose main aim is to teach a computer different gestures. Because there are very large sets of gestures, the number of methodologies to identify the set of gestures is also large. In this thesis, I have concentrated on the gestures are based on hands. The thesis is divided into two sections namely: Static mode and Dynamic. The Static mode concentrates on gestures based on still images and Dynamic mode concentrates on gestures based on image sequence.

As every hand gesture recognition system, the recognition paths has been divided into basically four parts for Static mode: Segmentation, Feature Extraction, Feature Selection and Classification. In the static mode the algorithm used are the GraphCut algorithm, Bacterial foraging optimization algorithm, Support vector machine, binary tree color quantization algorithm, block-based discrete cosine transform. The GraphCut algorithm uses the min-cut of a graph to separate the non-hand pixels from the hand pixels. The min-cut of the graph is found out by the max-flow algorithm. The binary tree color quantization algorithm is used to cluster the pixels into required number of clusters. The BFO algorithm is used to find the optimum value of parameter that are either required to be maximized or minimized. The BFO is an evolutionary algorithm which is a reflection of the swamping behavior of the E. Coli bacteria. The algorithm is a non-linear form of optimization and the convergence of the algorithm is faster than the other evolutionary algorithms.

For Dynamic mode the path has been divided into four parts: Segmentation, Tracking, Feature Extraction, Vector Quantization and Classification. The Dynamic mode uses 150 frames of image data to trace the path of the hand and finds the most likely gesture. The hand isolation is done by use of Gaussian Mixture model. To make the system as fast as possible the tracking of hand was more preferred to be fast than accurate. So some amount of accuracy was sacrifice for the sake of performance. As the sequence of image is involved the Hidden Markov model was preferred method for the classification. The training of the HMM was done by the method described by Baum- Welch which is the maximization of the expected value

of the parameters of the HMM. The training was followed by the testing where an image sequence of 150 frames was passed to the system. The Viterbi algorithm was used for the testing purposes. The Viterbi algorithm finds the most like sequence of states for which that particular sequence of observation is taken out.

Keywords: Graph Cut, Clustering, SVM, BFO, DCT, Color Quantization, Gesture, GMM, Tracking, HMM, Viterbi, Baum, Welch, pixels

CHAPTER 1: INTRODUCTION

Gestures have a part of our life from the dawn of civilization. One can convey a lot of emotions by just a single gesture. The study of gesture recognition is carried out a wide range of community. But mostly the field of gesture recognition belong to department of computer science. The main aim of the gesture recognition is to teach a machine to learn different gesture made by humans and to interpret the same. The possible location for origin of a gesture can be from any part of our body, but the most of the gesture studies that are carried out involve the gesture that have originated from the hand or the face. The reason because that most normal people convey the message through the hand movements or the facial expressions. The abundant available of dataset for different facial expression of the postures and movement of hand, attracts a lot of researcher to carry out their work in this field.

The entire point of Gesture recognition systems is to make a sensible conversation between a human and a machine possible. We humans as creature have always tried to make our life as comfortable as possible and the history stands proof that the introduction of machine has led to better standard of living. The application of hand gesture recognition is can be found in home appliances, professional photography, controlling a robot etc.

A. Gesture types

- Offline mode: The data that are collected from the image are interpreted after some period of time
- Online mode: This a light weight purpose method. Here the manipulation on image is directly done at the time of capture

B. Input Devices

- Wired gloves – This input method is used when a very high accuracy is required. The wired glove can detect the small change in movement of hand very accurately
- Depth- aware cameras – This input device is used when the 3-D representation of image is required.

- Stereo cameras – This input device uses two cameras to approximate the 3-D representation of image. Whenever there is need of depth – map this device is used.
- Controller-based gesture – This input device are the extension to main system. A specialized software is designed to capture the motion or the activities of the device.
- Single Camera – The device which feature in most work. The image obtained is 2-D. Further computation is done on the image to gather the required data.

CHAPTER 2: STATIC HAND GESTURE

INTRODUCTION

The static mode of Hand Gesture Recognition is based on gesture that features in single image. The gesture recognition path is divided into four parts:

- Hand Segmentation
- Feature Extraction
- Feature Selection
- Classification

The main motive behind hand segmentation is to isolate the pixels that belong to the hand from the surrounding. There are many novel techniques available but this thesis describes ITERATIVE GRAPH CUT [1] as a method to achieve the isolation of hand pixels from surrounding pixels.

Feature Extraction helps in extracting the key features that describe about the condition of the hand. By the condition of hand it is meant the characteristics of the particular hand like number of fingers raised, thickness of finger, complexity of hand etc.

Feature Selection helps finding the subset of the feature set which contribute the same information as the superset but without the noisy components. This reduces the number of features and thus increases computation speed.

Classification is a statistical method to classify the input vectors of features to several different classes. Rigorous mathematical analysis is required to carry-out such classification and this thesis describes one of the method "SUPPORT VECTOR MACHINE" [2] as a means to classify the inputs vectors.

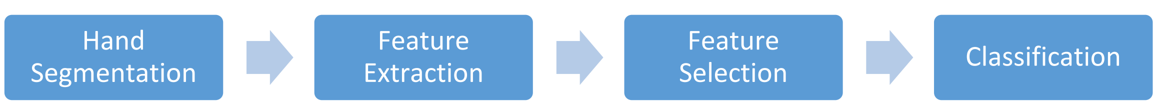

Figure 1.1: Path of Static mode of Hand Gesture Recognition

HAND SEGMENTATION

A. Iterative Graph Cut

Iterative Graph Cut carries out successive segmentation of image by using the min-cut of graph. This type of algorithm extends the GraphCut [1] algorithm to the case of color image. These development increases the popularity of the GraphCut algorithm which is mainly is applied in finding the min-cuts of the graph.

B. Summary

- The user is suggested to place his hand in the region of a marked rectangle. The region inside the rectangle is labelled as “Unknown” and the region outside the rectangle is labelled as “non-hand”.
- The initial segmentation that is created by the system is where all the pixels which has not been classified are inserted into a hand class and all the pixels that are known to belong to non-hand are inserted into the non-hand class.
- Orchard-Bauman [3] clustering algorithm is used to create the initial hand and non-hand Gaussian Mixture Models (GMM).
- In the hand class, every pixel is given a value that signifies to be the most probable Gaussian component in the hand class. The similar method is also followed for the non-hand class.
- The old values of GMMs are deleted and new values of GMMs are determined from the pixel data that were determined from the preceding set.
- The construction of graph is done based on the pixels and the GraphCut algorithm is run to determine a new possible hand and non-hand classification of pixels.
- The algorithm continues until a satisfactory result is not reached.

C. Data Structures

In order to run the GraphCut algorithm, four types of information is required for each and every pixel. Each type has its own array for storing the data. The size of the array is equal to the size of the image.

- Color(z) – a RGB value
- Trimap – this denotes the initial segmentation which has three possible values such as, either “TrimapNonhand”, “Trimaphand” or “TrimapUnknown”
- Matte(α) – this denotes the initial “hard segmentation” which can have either value “MatteNonhand” or “Mattehand” [4]
- Component Index (k) – this denotes a integer in the range of 1 to K

Every component, the values that are stored are:

- μ - this denotes the mean which a RGB value
- Σ^{-1} - this denotes covariance matrix’s inverse
- | Σ | - this denotes covariance matrix’s determinant
- π – this denotes a weight of component which is a real.

D. Initialization

- In step 1, the trimap is initialized by constructing a rectangle near the region which is assumed to be hand. For the pixel to be initialized as TrimapUnknown the pixel needs to be inside the rectangle otherwise it will initialized as TripmapNonhand This is the starting information that is passed into the algorithm.
- In step 2, the matte is given an initial value of Mattehand for all the pixels that belong to TrimapUnknown set and of MatteNonhand for all the pixels that belong to the TrimapNonhand set.
- In step 3, The K components of GMM are created based on the given matte value. A total number of 2K components are created. K clusters share the region. For each cluster the Gaussian components are intialized. In order to ensure a good separation between foreground and background, low variance Gaussian components are generated.

Thus it is required to find tight, well separated clusters. The color quantization technique described by Orchard-Bauman is used to achieve the above requirement.

E. Learning GMM components.

As the algorithm iterates, the matte will modify. During this process the transition of pixels between MatteForeground [4] and MatteBackground is observed. So in order to reflect the new foreground and background the GMMs need to be updated. One possible way to do so will be do re-run the clustering algorithm. But all clustering algorithm are slow. So instead, the GraphCut suggests to perform an incremental cluster update to make the process faster.

The incremental Gaussian Clustering algorithm [5] consists of steps:

- The first step to assign values to the pixels in the Mattehand. The value assigned is the hand GMM which has the largest probability of getting the color of the pixel.
- After clustering, the current GMMs are discarded and new GMMs are computed for each Matte index pair.

F. Performing GraphCut

The next step is to build a graph for the GraphCut algorithm.To construct a graph every pixel is denoted by a node. Then two separate node are also created for the hand node and non-hand node.The two unique node are connected by two types of links.One type is called A-links which is responsible for connecting the pixels that are in the near(mostly 8)-neighborhood. These links can be viewed as penalty for having a separating boundary between them. The system requires the penalty to be high where the gradient is low and vice-versa. The link value is kept constant the entire time.

B-links connect each pixel to foreground and background nodes. The link weight signifies the probability of the pixel belonging to the foreground or to the background. The updating of GMM and probabilities take place accordingly.

The link weight for A-links is given by [4]:

$$A(x, y) = \frac{\gamma}{d(x, y)} e^{-\beta||z_x - z_y||^2}$$

Equation 1.1: A-link weight

Where z_x is the color of pixel ‘x’ and γ=50 and d(x, y) is the Euclidean distance and

$$\beta = \frac{1}{2 < \left|\left|z_x - z_y\right|\right|^2 >}$$

Equation 1.2: Beta value

There are only two types of links for each and every pixel.

Table 1.1: Pixel Weight Assignment

Pixel Type	Background T-link	Foreground T-link
Trimaphand	0	N(x)
TrimapNonhand	N(x)	0
TrimapUnknown	D_{back}(x)	D_{fore}(x)

To make the pixel ‘x’ to be a member of either the hand or non-hand

$$N(x) = 8\gamma + 1$$

Equation 1.3: N(x) value

D_{hand} and $D_{non\text{-}hand}$ are given as follows for pixel x:

$$D(x) = -log \sum_{i=1}^{K} \pi_i \frac{1}{\sqrt{|\Sigma_i|}} e^{(-\frac{1}{2}|z_x - \mu_i|^T \Sigma_i^{-1} |z_x - \mu_i|)}$$

Equation 1.4: D(x) value

G. Color Clustering Details

Let say we have to set K clusters which will describe the hand GMM:

- Start with the set M_1 =Trimaphand U TrimapUnknown
- μ_1 which is the mean value of M_1 is calculated and Σ_1 which is the covariance matrix of M_1 is also calculated [3].

- For i=2 to K do
  - a set M_n which has the largest eigenvalues is searched and the corresponding eigenvector e_n is stored
  - then the next step is to divide M_n into two sets, $M_i = \{x \in M_n: e_n^T z_n < e_n^T \mu_n\}$
  - then values of μ_n^*, Σ_n^*, μ_i and Σ_i are calculated.

H. Results

Figure 1.2: Hand Segmentation Result

BACTERIAL FORAGING OPTIMIZATION

A. BFO Algorithm

In what follows, Bacterial Foraging Optimization (BFO) is briefly outlined step by step [6]:

a. Initialing parameters p, Sr, N_c, N_s, N_{re}, Ned, P_{ed}, c(i) (i=1, 2, …, S), $Θ_i$ where:
   - p: Dimension of the search space
   - S: The number of bacteria
   - N_c: Chemotaxis steps
   - N_{re}: Reproduction steps
   - N_{ed}: Elimination and Dispersal steps
   - c(i): The run-length unit during each run or tumble

b. Elimination-dispersal loop: l=l+1.

c. Reproduction loop: k=k+1.

d. Chemotaxis loop: j=j+1.
   1. For i=l=1, 2,….,S take a chemotaxis step for bacteria I as follows.
   2. Compute the fitness function J(i,j,k,l).
   3. Let J_{last} = J(i,j,k,l) to save the value since we may find better value via run.
   4. Tumble: Generate a random vector d(i) Є R^n.
   5. Move: Update of the position.
   6. Compute the fitness function J(I,j+1,k,l) with $Θ^i$(I,j+1,k,l).
   7. Swim:
      I. Let m = 0 (counter for swim length).
      II. While m< Ns (if have not climbed down too long).Let m = m+1.
      If J(i,j+1,k,l) < J_{last}, let J_{last} = J(i,j+1,k,l).
      Then another step of size c(i) in this same direction will be taken and use the new generate $Θ_i$(I,j+1,k,l) to compute the J_{last} = J(I,j+1,k,l).
      Else let m = N_s.
   8. Go to the next bacterium (i+1): if i is not equal to S go to step 2 to process the next bacteria.

e. if j < N_c, go to step c. In this case, continue chemotaxis since the life of the bacteria is not over.

f. Reproduction:

1. For the given k and l, and for each i=1,2,….,S, let J_{health} be the health of the i_{th} bacteria. The bacteria are sorted in the order of ascending values.

$$J_{health} = \sum_{i=1}^{Nc+1} J(i,j,k,l)$$

2. The bacteria with the highest J_{health} values also dominated die, and the other non-dominated bacteria with best J_{health} values reproduce. The number of the dead individuals is no more than Sr. The copy the best bacteria in order of keeping the group number unchanged.

g. If $k < N_{re}$ go to step b. When this state is reached the it means the number of reproduction state is less than the required number, so a next generation of chemotaxis step is done.

h. Eliminate-dispersal: for i=1,2,…S the bacteria is dispersed with probability P_{ed} which results in the number of bacteria being constant in the population. This is done by moving the bacteria to a random location. If l< Ned loop back to step b, or end otherwise.

B. Flow Chart

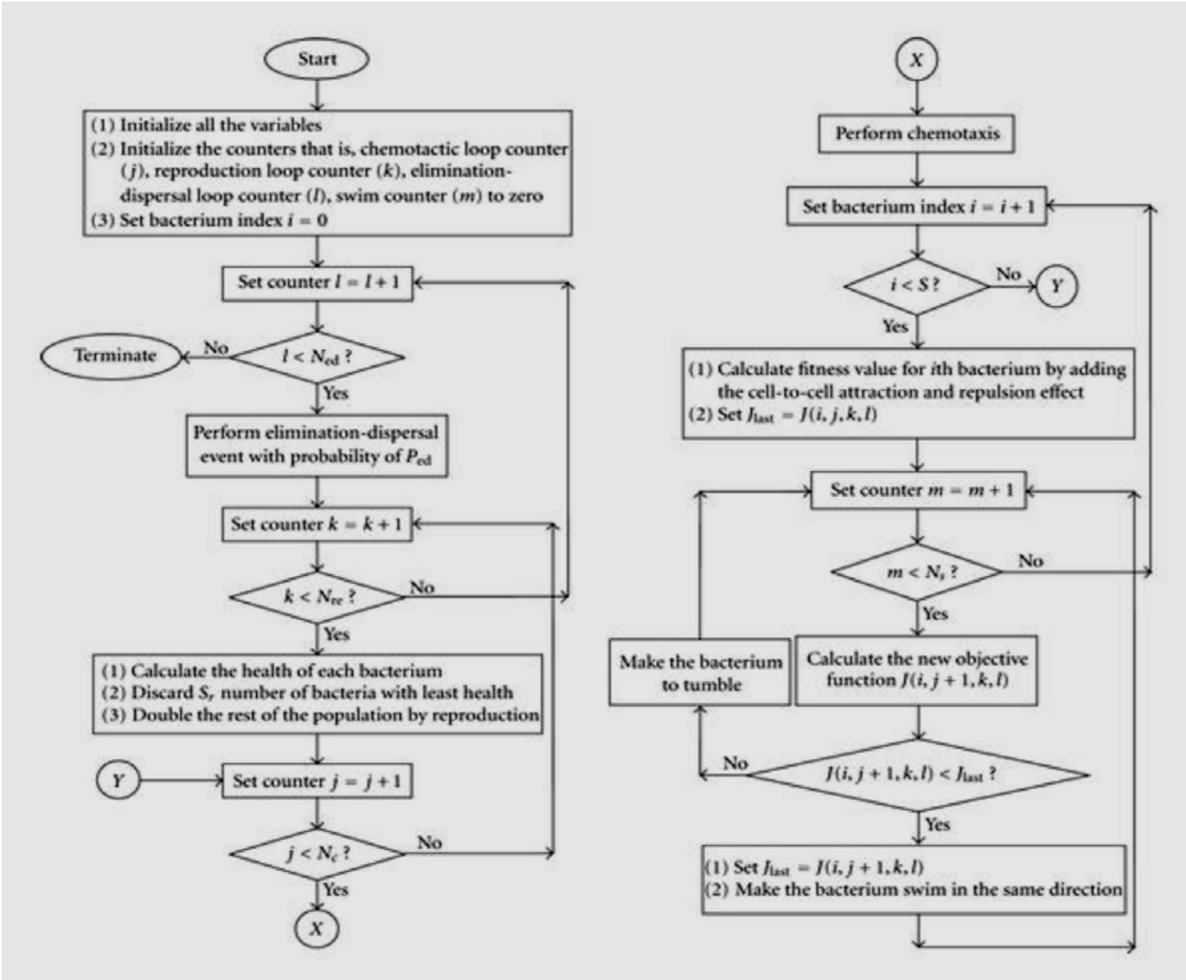

Figure 1.3: Flow Chart of BFO Algorithm

BLOCK-BASED DISCRETE COSINE TRANSFORM

To extract the feature by applying Block-based DCT [7], the image is first split into blocks of equal size. This ensure the isolation of most important features. The most important feature regarding the hand is stored in the DC component or the low-frequency component. The high frequency components store the finer details like width, thickness etc. This component is highly vulnerable to change in pose and expression.

Mathematically, the 2D-DCT of an image is given by [7]:

$$F(u,v) = \alpha(u)\alpha(v)\sum_{x=1}^{M-1}\sum_{y=1}^{N-1}\cos(\frac{\pi u}{2M}(2x+1))\cos(\frac{\pi v}{2N}(2y+1))$$

$$\alpha(u),\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & u,v = 0 \\ \sqrt{\frac{2}{N}} & u,v \neq 0 \end{cases}$$

Equation 1.5: Discrete Cosine Transform

Where f(x, y) is the intensity of the pixel at coordinates (x, y), u varies from 0 to M-1, and v varies from 0 to N-1, where M X N is the size of the image.

In this thesis, blocks of 8X8 is preferred as a measurement to divide the image. Zeros are padded along the rows and column if the if the size is not multiple of 8.

The choice of size 8 is done because

- No compromise is done will collecting the data from image.
- Variation of image content is graceful.
- It is best suited for hardware implementation.

FEATURE SELECTION

Feature selection also known as variable selection is process of selection of most relevant feature subset. The assumption behind the implementation of the technique is that every feature set contains some reductant or noisy feature components which do not contribute towards the overall information. This in turn increases computational overload without any benefit. So it becomes a requirement to remove such noisy feature components.

Feature selection techniques provide three main benefits

- The computation becomes more faster
- Less chance of over-fitting the dataset.
- Better model interpretability

A. BFO based feature selection

Bacteria Representation

The position of the bacteria represents the possible solution to the selection of feature subset with minimum reductant components. If 'l' is the length of feature set extracted by DCT, then the number of dimension of search space is also 'l'. For each dimension of search space the position of bacteria can take value 1 or 0. Here 1 means that the bacteria is selected and 0 means it is not. In each iteration the bacteria moves to a new random location. Position of i^{th} bacteria in j^{th} chemo taxis and k^{th} reproduction step is defined as:

$$\theta^i = f_1 f_2 \dots \dots f_l$$

Equation 1.6: Possible solution

Fitness Function

In each iteration the position of the bacteria are evaluated and value of fitness is returned by a fitness function. Let $c_1, c_2, \dots, c_p$ represent the classes of images and $n_1, n_2, \dots, n_p$ represent the number of images in the respective classes. Let $m_1, m_2, \dots, m_p$ represent the mean of the respective class and m_0 represent the grand mean. m_i can be calculated as:

$$m_i = \frac{1}{n_i} \sum_{j=1}^{n_i} w_j^{(i)}$$

Equation 1.7: Mean of each class

$$m_0 = \frac{1}{n} \sum_{i=1}^{p} n_i m_i$$

Equation 1.8: Grand Mean

Where n is the total number of images.

Thus the fitness function is given as:

$$F = \sqrt{\sum_{i=1}^{p} (m_i - m_0)^T (m_i - m_0)}$$

Equation 1.9: Fitness Function

SUPPORT VECTOR MACHINE

A. Introduction

Support Vector Machine [2] is a classification problem where the algorithm formulates a set of hyperplane which separate two or more classes. Thus it can be used for classification and regression analysis. One can intuitively say that a good classifier is one which maximizes the separation between the two nearest training data points that belong to two different class. This ensure that the number of error committed by the classifier will be minimum.

The original research on SVM focused on the linearly separable dataset, but most of the practical application involve non-linearly separable dataset. For this reason, it was proposed to map the lower-dimensional data points into a higher-dimensional space via a "kernel function" ϕ(x, y). The hyperplane in defined as collection of points in higher-dimension such that the dot product with a vector in that space in constant.

$$\sum_i a_i \varphi(x, x_i) = constant$$

To obtain an optimum set of values of 'a', L(a) must be maximized

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j \varphi(x_i, x_j)$$

Equation 1.10: Maximization by Lagrangian Multiplier

Subject to constraints $a_i \geq 0$ and $\sum_{i=1}^{n} a_i y_i = 0$

B. BFO based Optimization

To obtain an optimum value for C, the Regularization parameter and ‘a’ the lagrangian multipliers, BFO is employed.

Bacteria Representation

The position of bacteria [8] represents one possible solution to the C and a. The dimension of search space is n which is same as the dimension of DCT feature set after feature selection. In each dimension, the value of C is a floating-point value and the value of a_i is restricted to $0 \leq a_i \leq C$

Fitness Function

In each iteration a new value of ‘a’ and C are generated and evaluated with the fitness function to determine the goodness of the new value. The fitness function is given by:

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \varphi(x_i, x_j)$$

Equation 1.11: Fitness function for SVM

MULTI-CLASS SVM

In practical life a set of data points rarely belong to only two sets of class. Here in this project we have employed one of many multi-class SVM [9] method to achieve our goal. Although SVMs were originally designed as binary classifiers, approaches that address a multi-class problem as a single “all-together” optimization problem exist, but are computationally much more expensive than solving several binary problems

A. One v/s One SVM

This algorithm constructs $\frac{N(N-1)}{2}$ two-class classifiers [9], using all the binary pair-wise combinations of the N classes. Each classifier is trained using the samples of the first class as positive examples and the samples of the second class as negative examples. To combine these classifiers, the Max Wins algorithm is adopted. It finds the resultant class by choosing the class voted by the majority of the classifiers. The number of samples used for training of each one of the one v/s one classifiers is smaller, since only samples from two of all N classes are taken in consideration.

- Advantage - The lower number of samples causes smaller nonlinearity, resulting in shorter training times.
- Disadvantage - every test sample has to be presented to large number of classifiers$\frac{N(N-1)}{2}$. This results in slower testing, especially when the number of the classes in the problem is big.

PERFORMANCE AND RESULTS

The computer was trained to determine number of finger raised in each image. The database that were used were Microsoft Kinect Hand Database, Sebastien Marcel Static Hand Posture Database and Lab database created by own.

Table 1.2: Performance Analysis

Database	Training Images	Testing Images	Correctly Recognized	Accuracy (%)
Microsoft Kinect	100	50	46	92.00
Sebastian Marcel	100	50	47	94.00
Lab	50	25	24	96.00

Some test results:

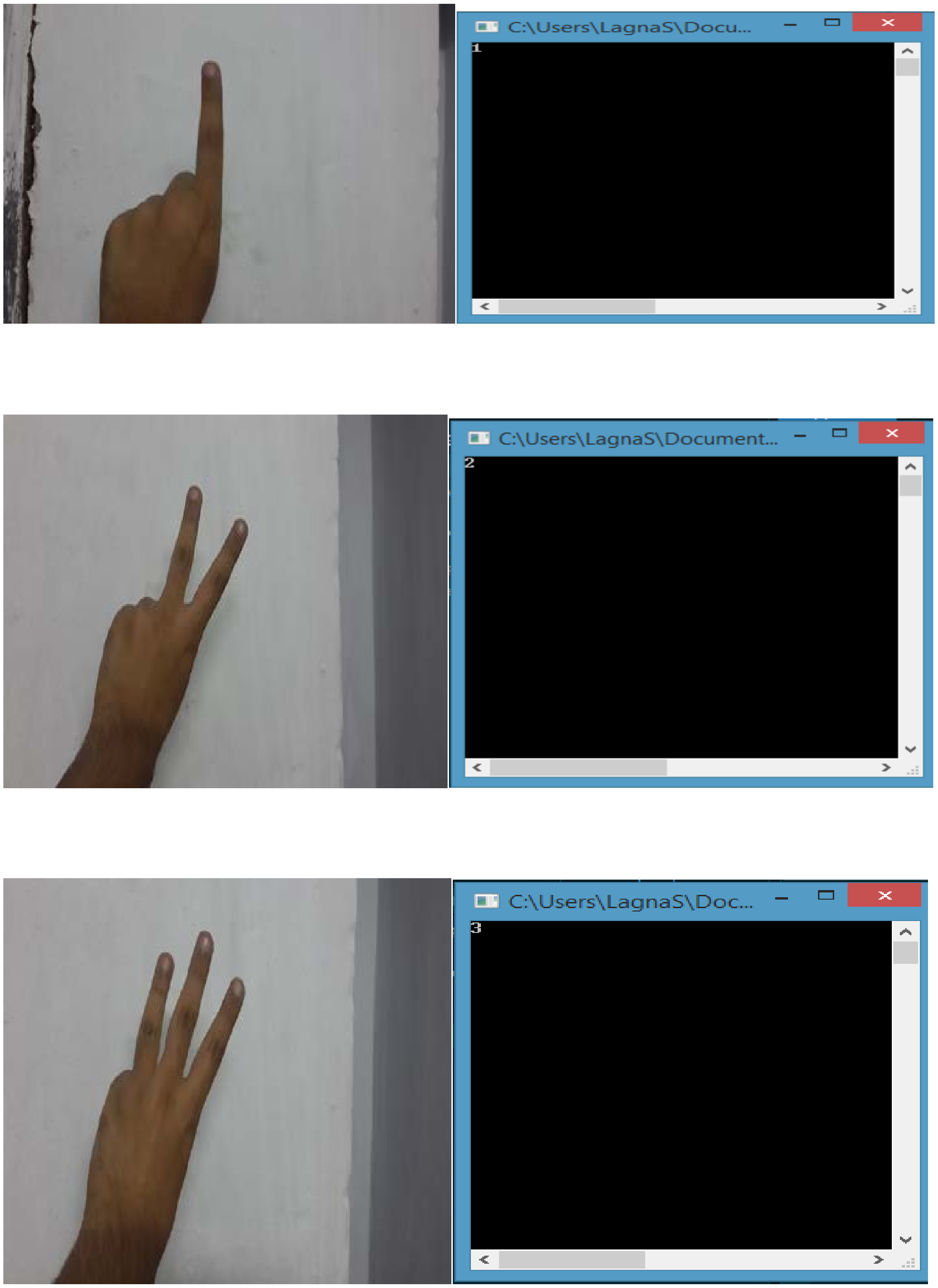

Figure 1.4: Recognition results for a)1 b)2 c)3

SUMMARY

The static hand gesture recognition is designed for interpreting a single image. The image data is passed to the GraphCut algorithm to obtain a most likely segmentation of hand from the background. The next step was to extract the features from the image. So a block-based DCT was employed to extract the suitable feature from the image. The feature set was further reduced by a feature selection algorithm which was back by BFO algorithm. The reduced set was passed to multi-class classifier to train the classifier. Same steps were followed for the test image and fed to the classifier to find the accuracy of the system.

CHAPTER 3: DYNAMIC HAND GESTURE

INTRODUCTION

In Dynamic mode of Hand Gesture Recognition, the gestures are recognized from the motion and posture of the hand. This enables a system designer to use the hand motion to recognize different sign languages. The present application and all possible future application of this mode is very large. For instance, one can use hand gestures to control the home appliances, people with disabilities can be benefitted with such technology, and hand gestures find their application in video conferencing like controlling a presentation.

The dynamic mode has been divided into 4 parts:

- Skin Color detection – In order to begin the recognition path, one must recognize the hand in the image sequence. Skin Color detection is a proven method that isolates the skin pixels from the surrounding pixels. In order to differentiate the hand pixels from face pixels, this project has restricted the motion of hand to certain region in a single frame.
- Hand Tracking – To detect the path of motion of hand, the hand needs to be tracked. The hand as a whole can be represented by the centroid of all hand pixels that form the hand in motion. In order to facilitate a Real-Time system, the hand tracking must be very fast. So some accuracy must be sacrificed for speed. For this "compressive tracking" is used.
- Feature Extraction and Vector Quantization – After the tracked points of hand are stored, the data must be analyzed to make some sense out of it. The feature extraction technique employ the position, velocity and angle characteristics of motion to define different form of motion. The vector quantization assigned code-word to different features which will be in turn fed to the classification block.
- Classification – Because the sequence of image is involved and this sequence of image will represent different states of hand in motion. So use of Hidden Markov Models is very optimal for this case. Baum-Welch Algorithm is used to train the classifier and Viterbi Path algorithm is used to determine the most probable sequence of state.

Figure 2.1: Recognition Path for Dynamic mode

SKIN COLOR DECTECTION

The YC_bC_r color space is used to detection of skin pixels. The skin color van be different based on ethnicity, climate etc. So a model needs to be constructed which can be a best estimate of skin color. For this purpose, this thesis describes the Gaussian Mixture Model. The models is formulated based on properties of Human Skin Color using the Extended Yale Face Database B (http://vision.ucsd.edu/content/extended-yale-face-database-b-b) [10].

A. Gaussian Mixture Model

Every skin pixel may be seen as a part of a larger population P, which again can been a mixture of smaller finite number, p, of sub-population P_1, P_2,…., P_p in some proportion a_1,a_2,…. , a_p respectively, where

$$\sum_{i=1}^{p} a_i = 1 \quad and \quad a_i \geq 0$$

So the probability density function (p.d.f.) of an observation 'x' (of dimension 'n') in the Gaussian Mixture model is

$$\begin{aligned} p(x;\varphi) &= \sum_{i=1}^{p} a_i p_i(x;\theta) \\ &= \sum_{i=1}^{p} a_i p(x|i;\theta) \\ &= \sum_{i=1}^{p} a_i \frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} \exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)) \end{aligned}$$

Equation 2.1: Gaussian Mixture Model

Where $p_i(x;\theta)$ is the p.d.f of P_i and θ is an unknown parameter.

For multivariate Gaussian components, θ contains the components of μ_i which is the mean vector and the distinct components of Σ_i which is the covariance matrices.

The vector

$$\varphi = (\pi^T, \theta^T)^T$$

Is estimated by the use of EM Algorithm.

B. EM Algorithm

The Gaussian Mixture Model contains parameters like π_i, a_i and Σ_i which can be adjusted at any time. So the negative log-likelihood for the data is described by:

$$E = -\ln L = -\sum_{j=1}^{N} \ln \mathrm{p}(x_j) = -\sum_{j=1}^{N}\sum_{i=1}^{p} a_i p(x_j|i)$$

Equation 2.2: Negative Log-Likelihood

This function can be considered as an error function. So in order to minimize E it is equivalent to maximize L.

The EM algorithm [11] starts by making some initial assumptions regarding the parameters of the Gaussian Mixture Model, which is known as “old” values. The new values are evaluated using the following equations.

The change in the error function when the old parameter values are replaced by new values is given by:

$$\Delta^{t+1} = E^{t+1} - E^{t} = -\sum_{j=1}^{N} ln(\frac{p^{t+1}(x_j)}{p^{t}(x_j)})$$

Equation 2.3: Change in error function

Where $p^{t+1}(x)$ is the p.d.f found out by using new parameter values and $p^t(x)$ is the p.d.f found out using old parameter values.

To find the maximum value we need to set $derivatives\ of\ \Delta^{t+1} = 0$, thus we obtain the equation for the new values of parameter.

$$\mu_i^{t+1} = \frac{\sum_{j=1}^{N} p^t(i|x_j)x_j}{\sum_{j=1}^{N} p^t(i|x_j)}$$

Equation 2.4: New Mean vector

$$\Sigma_i^{t+1} = \frac{\sum_{j=1}^{N} p^t\left(i\middle|x_j\right)||x_j - \mu_i^t||^2}{\sum_{j=1}^{N} p^t(i|x_j)}$$

Equation 2.5: New Covariance matrix

$$a_i^{t+1} = \frac{1}{n}\sum_{j=1}^{N} p^t(i|x_j)$$

Equation 2.6: New Weight vector

Where

$$p^t\left(i\middle|x_j\right) = \frac{p^t\left(x_j\middle|i\right)a_i^{t+1}}{p^t\left(x_j\middle|i\right)}$$

C. Results.

HAND TRACKING: COMPRESSED VECTORS

A. Random Projection

Let ‘A’ be a random matrix such that Aϵ $R^{p \times q}$ has row of unit length, maps data from the high dimension x ϵ R^q to a lower dimension y ϵ R^p

$$y = Rx$$

Equation 2 7: Projection

where p<<q.

According to Johnson-Lindenstrauss lemma [12], the distance between two points in a vector space can be mapped to a randomly selected subspace with very high probability with suitably high dimensions.

B. Random Measurement Matrix

A typical measurement matrix [12] is Gaussian matrix Aϵ $R^{p \times q}$ where a_{ij} ~ N(0,1). But this matrix is a dense matrix which will lead to high computational overload. So a sparse random measurement matrix is adopted where

$$a_{ij} = \sqrt{b} \times \begin{cases} 1, & with\ probability\ 1/2b \\ 0, & with\ probability\ 1-1/b \\ \quad -1, & with\ probability\ 1/2b \end{cases}$$

Equation 2.8: Entries for random measurement matrix

Setting b=a/4 makes a very sparse matrix

C. Proposed Algorithm

In each tracking frame, some positive samples are taken at the target location and some negative samples are taken at the location far away from the target center. To facilitate the prediction of the hand in the next frame, some samples are drawn at the hand's location and a classifier is employed to find the most suitable sample.

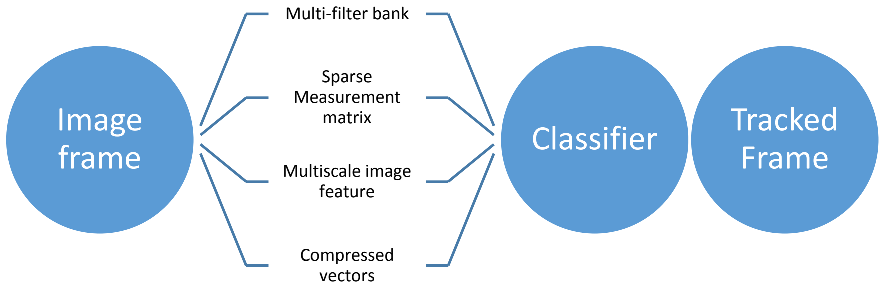

Figure 2.2: Tacking flow

Efficient Dimensionality Reduction

There is a problem of scaling which can be dealt by use of convolution of each sample 's' with a set of rectangular filters at multiple [13] scales $\{m_{11}, m_{12}, \ldots, m_{wh}\}$, which is defined as:

$$m_{i,j}(x,y) = \begin{cases} 1, 1 \le x \le i, 1 \le y \le j \\ 0, \qquad\qquad otherwise \end{cases}$$

Where i and j are the width of the rectangular filter.

Classifier construction and update

For every sample x, there exists a low-dimensional representation y.

It has been assumed that the 'y' are independently distributed and so they are modelled with a naïve Bayes classifier.

$$H(y) = \sum_{t=1}^{n} \log(\frac{p(x_i|a = 1)}{p(x_i|a = 0)})$$

Equation 2.9: Bayesian Classifier

Where it has been assumed that $p(x_i|a = 1) = p(x_i|a = 0)\ and\ a\ \epsilon\ \{0,1\}$

The conditional distribution $p(x_i|a = 1)$ and $p(x_i|a = 0)$ have been assumed be Gaussian with parameter (μ_i^1, σ_i^1, μ_i^0, σ_i^0) where

$$p(x_i|a = 1) \sim N(\mu_i^1, \sigma_i^1),\ \ p(x_i|a = 0) \sim N\left(\mu_i^0, \sigma_i^0\right)$$

The parameters are incremented as

$$\mu_i^1 \leftarrow \lambda\mu_i^1 + (1 - \lambda)\mu^1$$

$$\sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1 - \lambda)(\sigma^1)^2 + \lambda(1 - \lambda)(\mu_i^1 - \mu^1)^2}$$

Equation 2.10: Parameter Update

Where $\lambda > 0\ is\ a\ learning\ parameter$

$$\sigma^1 = \sqrt{\frac{1}{n} \sum_{k=0|y=1}^{n-1} (x_i(k) - \mu^1)^2}$$

$$\mu^1 = \frac{1}{n} \sum_{k=0|y=1}^{n-1} x_i(k)$$

D. Results

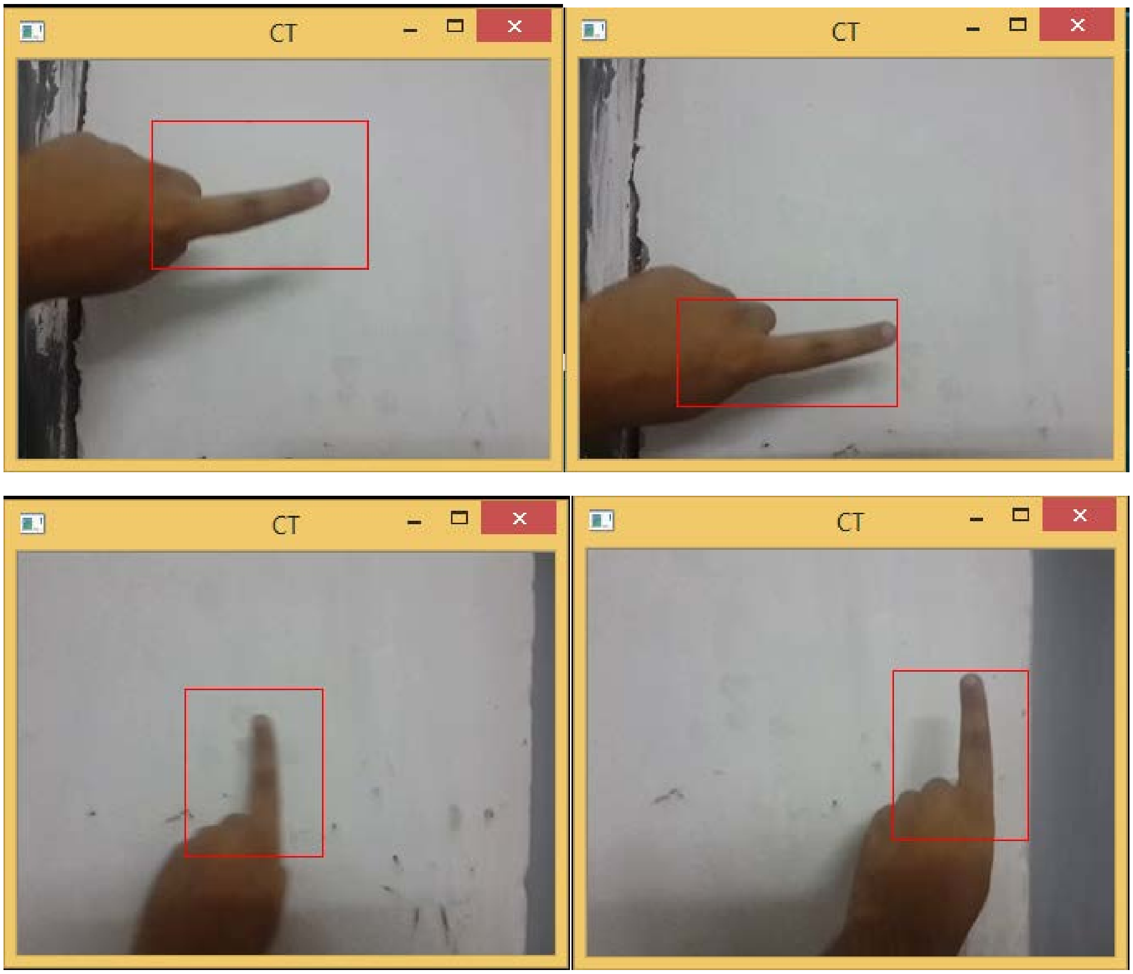

Figure 2.3: Tracking Results a) top-down movement b) sideways movement

FEATURE EXTRACTION AND QUANTIZATION

In a motion-based gesture recognition, there are 3 basic types of features

- Location feature
- Orientation feature
- Velocity feature

Let (X_1, Y_1) be the initial location of centroid of hand

Let (X_t, Y_t) be the current location of centroid of hand

Let (X_{t+1}, Y_{t+1}) be the location of centroid of hand in next frame

The location feature is given by:

$$F_l = \sqrt{(X_t - X_1)^2 + (Y_t - Y_1)^2}$$

The orientation feature is given by

$$F_o = \tan^{-1}\frac{Y_t - Y_1}{X_t - X_1}$$

The velocity feature is given by

$$F_v = \sqrt{(X_{t+1} - X_t)^2 + (Y_{t+1} - Y_t)^2}$$

The orientation feature is quantized by dividing it into 12 parts each of 30^o .

CLASSIFATION: HIDDEN MARKOV MODEL

HMM [14] is a statistical model which is use to model stochastic processes. A stochastic process is process in which a random sequence of outcomes are generated based on some probabilities. A HMM can be described by a triple (X, Y, Z):

- A process transitions through several state s_i ϵ S where S is the set of all states. Total number of state is N
- Every state s_i is associated with a probability Z_i so Z_i =P(s_i)
- The probability that the process at state s_i will make a transition to state s_j given by X_{ij} where i,j ϵ [1,N]. X is knows as transition matrix and a dimension of N x N. The sum of each row X must be equal to 1.
- Based on transition of states and probability, an observation o_i at instant t. The set of all observation is given by 'O' and the total time period is the length of the pure path, T.
- Let the symbols that are observed be M={m_1,m_2,…. m_v} where V is the total number of symbols

- The probability that a symbol m_j will be emitted when the process is in the state s_i is given by Y_{ij}. Y is the observation matrix. The sum of all values in a row must be equal to one.

A. Initialization

The number of state the process transitions through in this thesis is equal to 6. The number of symbols emitted is equal to 4.

The initialization of transition matrix [15] is dependent on the duration time t for each symbol. So

$$t = \frac{T}{N}$$

Equation 2.11: Duration time

Where T denotes the length of the pure path and N denotes the number of states.

$$X = \begin{bmatrix} x_{ii} & 1-x_{ii} & 0 & 0 & 0 & 0 \\ 0 & x_{ii} & 1-x_{ii} & 0 & 0 & 0 \\ 0 & 0 & x_{ii} & 1-x_{ii} & 0 & 0 \\ 0 & 0 & 0 & x_{ii} & 1-x_{ii} & 0 \\ 0 & 0 & 0 & 0 & x_{ii} & 1-x_{ii} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 2.12: Initial Transition Matrix

$$x_{ii} = 1 - 1/t$$

For the initialization of observation matrix

$$Y_{ij} = 1/V$$

$$Y = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

Equation 2.13: Initial Observation Matrix

The initial value of Z vector:

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Equation 2.14: Initial Probability of states

B. Training: Baum-Welch Algorithm

Baum-Welch Algorithm [16] uses forward and backward algorithm to determine the unknown parameters of the HMM. The basic assumption behind the working of the algorithm is that the probability distribution of ith state is only dependent on the probability distribution of (i-1)th state.

Forward Algorithm

Let $a_i(i) = P(O_1 = o_1, O_2 = o_2, O_3 = o_3, \dots., O_t = o_t, S_t = i|\theta)$

$$a_i(1) = z_i y_i(o_1)$$

$$a_j(t+1) = y_j(o_{t+1}) \sum_{i=1}^{N} a_i(t) x_{ij}$$

Equation 2.15: Forward Algorithm

Backward Algorithm

Let $b_i(i) = P(O_{t+1} = o_{t+1}, O_t = o_t, O_{t-1} = o_{t-1}, \dots., O_T = o_T|S_t = i, \theta)$

$$b_i(T) = 1$$

$$b_i(t) = \sum_{j=1}^{N} b_j(t+1) x_{ij}\, y_j(o_{t+1})$$

Equation 2.16: Backward Algorithm

Update

$$c_i(t) = P(S_t = i | O, \theta) = \frac{a_i(t) b_i(t)}{\sum_{j=1}^{N} a_j(t) b_j(t)}$$

$$d_{ij}(t) = P(S_t = i, S_{t+1} = j | O, \theta) = \frac{a_i(t) x_{ij} b_j(t+1) y_j(o_{t+1})}{\sum_{k=1}^{N} \sum_{l=1}^{N} a_k(t) x_{kl} b_l(t+1) y_l(o_{t+1})}$$

$$= \frac{a_i(t) x_{ij} b_j(t+1) y_j(o_{t+1})}{\sum_{k=1}^{N} a_k(t) b_k(t)}$$

The new values are given by:

$$z_i^* = c_i(1)$$

$$x_{ij}^* = \frac{\sum_{t=1}^{T-1} d_{ij}(t)}{\sum_{t=1}^{T-1} c_i(t)}$$

$$y_i^*(m_k) = \frac{\sum_{t=1}^{T} 1_{o_t = m_k} c_i(t)}{\sum_{t=1}^{T} c_i(t)}$$

Where $1_{o_t = m_k} = \begin{cases} 1, if\ o_t = m_k \\ 0, otherwise \end{cases}$

Equation 2.17: Update

C. Most likely Sequence of State: Viterbi Path

In order to find the most likely hidden sequence from the sequence of observed output, Viterbi algorithm is used. Viterbi algorithm [14] is a dynamic programming algorithm. The most like sequence of hidden state is known as Viterbi path.

Algorithm

Let $P_{t,s}$ be the probability of most likely sequence of state that is the cause of 't' observations which has 's' as its final state

Then

$$P_{1,s} = y_s(o_1).z_s$$

$$P_{t,s} = \max_{x \epsilon S}(y_s(o_t)x_{st}P_{t-1,s})$$

Let Path(k,t) be the function that returns the state 'x' used to compute $P_{t,s}$ if t>1 or 's' if t=1

$$x_T = arg\max_{x \epsilon S} P_{t,x}$$

$$x_{t-1} = Path(x_t, t)$$

Equation 2.18: Viterbi Path

Performance

Table 2.1: Performance table

Symbol	**#Training Video**	**#Testing Video**	**#Correctly Recognized**	**Accuracy (%)**
'1'	5	10	10	100
'2'	5	10	8	80
'3'	5	10	7	70
'4'	5	10	7	70
'5'	5	10	8	80

SUMMARY

The dynamic hand gesture recognition system is designed for interpreting image sequence. In this thesis an image sequence of 150 frames is used to both train and test the system. The image sequence was fed to hand isolation block which will isolate the hand pixels in each frame. A Gaussian Mixture Model was formulated by using the Extended Yale face database. The isolated image were fed to hand tracking block which will track the movement of hand in each frame and tries to find path of movement. The feature extraction method extracts the path of movement from the image sequence. The code-words are assigned to different path so they will be fed to the HMM classifier in next step. The HMM is trained by using Baum-Welch algorithm and the image sequences are tested by Viterbi algorithm.

CHAPTER 4: CONCLUSION AND FUTURE WORK

A. CONCLUSION

By analyzing the results and performance of both static and dynamic mode, it was concluded that the static mode performs better in all aspect. But this result was expected because the dynamic mode is mathematically more complex and computationally is more intensive than the static mode. The dynamic mode requires a better class of hardware than the static mode and given that the hardware requirement is fulfilled, it can be possible to achieve better performance than the static mode.

The bottleneck analysis of the static mode showed that most of the computation time and resource was spent in the segmentation part with classifier following in the second place by a very small margin. The training period has not been taken into account because mostly it was one-time.

The bottleneck analysis of the dynamic part showed that the most of the computation time and resource was spent in the tracking part. This was intuitive because it requires processing the pixels of all the participating image frames and as expected it can be very intensive task.

The bottleneck problem in both the mode can be solved by upgrading to a better class of hardware like multi-core processors and GPUs. Recent advancement in VLSI has made possible for implementing multi-threading at hardware-level. The research in field of parallel computing has made possible for parallelizing many sequential algorithms. So a suggestion can be made to work on the parallel implementation of the above thesis in future.

B. FUTURE WORK

- To improve the segmentation algorithm for the static hand gesture recognition.
- To improve the feature selection algorithm for better selection of subset.
- To improve the hand isolation algorithm for better isolation of hand different with skin color, with wide range of brightness of image.

- To improve the hand tracking algorithm, so that a real-time system is possible.
- To work on a wide range of character and symbols for the dynamic hand segmentation.
- To formulate a parallel implementation of the system.

References

[1] Y. Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," in *Internation Conference on Computer Vision*, Vancouver, Canada, July 2001.

[2] B. Fei and J. Liu, "Binary Tree of SVM: A New Fast Multiclass Training and Classification Algorithm," *IEEE Transaction on neural networks,* vol. 17, no. 3, May 2006.

[3] M. T. ORCHARD and C. A. BOUMAN, "Color Quantization of Images," *IEEE Transactions on Signal Processing,* pp. 39, 12, 2677–2690, 1991.

[4] J. F. Talbot and X. Xu, "Implementing GrabCut," Utah, 2006.

[5] D. Ross, J. Lim, R. Lin and M.-H. Yang, "Incremental learning for robust visual tracking," *IJCV,* no. 77, p. 125–141, 2008.

[6] W. Tang and Q. Wu, "Bacterial foraging algorithm for dynamic environment," *Proceedings of IEEE Congress on Evolutionary Computation,* pp. 1324-1330, 2006.

[7] K. Manikantan, V. Govindarajan, V. S. Kiran and S. Ramachandran, "Face Recognition using Block-Based DCT Feature Extraction," *Journal of Advanced Computer Science and Technology,* vol. 1, no. 4, pp. 266-283, 2012.

[8] E. Zitzler and L. Thiele, "Multi-objective evolutionary algorithms," *IEEE trans,* pp. 257-271, 1999.

[9] J. Weston and C. Watkins, "Multi-class support vector," *Proceedings of ESANN99,* 1999.

[10] U. C. Vision, "Extended Yale Face Database B," [Online]. Available: http://vision.ucsd.edu/content/extended-yale-face-database-b-b. [Accessed January 2015].

[11] N. Soontranon, S. Aramvith and T. H. Chalidabhongse, "Improved Face and Hand Tracking for Sign Language Recognition," in *ITCC 2005*, April 2005.

[12] H. Li, C. Shen and Q. Shi, "Real-time visual tracking using compressive sensing," in *CVPR*, Providence, RI, 2011.

[13] D. Donoho, "Compressed sensing," *IEEE Trans. Information Theory,* p. 1289–1306, 2006.

[14] S. Marcel, O. Bernier, J. Viallet and D. Collobert, "Hand Gesture Recognition using Input-Output Hidden Markov Model," *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 456-460, 2000.

[15] N. Liu, B. C. Lovell, P. J. Kootsookos and R. I. A. Davis, "Understanding HMM Training For Video Gesture Recognition," in *IEEE TENCON*, Nov. 2004.

[16] N. Liu, B. C. Lovell, P. J. Kootsookos and R. I. A. Davis, "Understanding HMM Training For Video Gesture Recognition," *IEEE TENCON,* pp. 567-570, 2004.

[17] C. ROTHER, V. KOLMOGOROV and A. BLAKE, "Grabcut - interactive foreground extraction using iterated graph cuts," *ACM Siggraph,* 2004.

[18] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Transactions on Systems, MAN, and Cybernetics,* pp. 311-324, May 2007.

[19] S. Marcel, "Sébastien Marcel - Hand Posture and Gesture Datasets," [Online]. Available: http://www.idiap.ch/resource/gestures/. [Accessed August 2014].

[20] N. Liu, B. C. Lovell and P. J. Kootsookos, "Evaluation of HMM Training Algorithms for Letter Hand Gesture Recognition," *IEEE International Symposium on Signal Processing and Information Technology,* pp. 648-651, December 2003.

[21] X. Dezou, "A Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG," *ICPR, 18th International Conference on Pattern Recognition,* pp. 519-522, 2006.

[22] K. Deb, P. A., A. S. and M. T., "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans,* pp. 182-197, 2002.

[23] S. Das, S. Dasgupta, A. Biswas, A. Abraham and A. Konar, "On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm," *IEEE Trans. Syst. Humans 39,* pp. 670-679, 2009.

[24] Y. CHUANG, B. CURLESS, D. H. SALESIN and R. SZELISKI, "A Bayesian approach to digital matting," *IEEE Conf. on Computer Vision and Pattern Recognition, IEEE Computer Society,* vol. II, p. 264–271, 2001.

[25] M. R. Cambridge, "MSRC-12 Kinect gesture data set Microsoft Research Cambridge," Microsoft, [Online]. Available: http://research.microsoft.com/en-us/um/cambridge/projects/msrc12/. [Accessed August 2014].

[26] P. Baggenstoss, "A modified Baum-Welch Algorithm for hidden Markov models with multiple observation space," *Speec and Audio Processing,* vol. 9, no. 4, pp. 411-416, 2001.