Gesture based Character Recognition

Lalit Mohan Pradhan

Roll No. 213CS1148

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Gesture based Character Recognition

Dissertation submitted in

May 2015

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

Lalit Mohan Pradhan

(Roll No. 213CS1148)

under the supervision of

Prof. Banshidhar Majhi

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

May 25, 2015

Certificate

This is to certify that the work in the thesis entitled ***Gesture based Character Recognition*** by ***Lalit Mohan Pradhan***, bearing roll number **213CS1148**, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering Department.* Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

prof. Banshidhar Majhi
Computer Science Department

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Banshidhar Majhi, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Character Recognition and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I thank our H.O.D. Prof. Santanu Kumar Rath, Prof. Ruchira Naskar and prof. Bidyut Kumar Patra for their constant support in my thesis work. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would also like to thank all faculty members, PhD scholars, my seniors and juniors and all colleagues to provide me their regular suggestions and encouragements during the whole work.

At last but not the least I am in debt to my family to support me regularly during my hard times.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Lalit Mohan Pradhan

Abstract

Gesture is rudimentary movements of a human body part, which depicting the important movement of an individual. It is high significance for designing efficient human-computer interface. An proposed method for Recognition of character (English alphabets) from gesture i.e gesture is performed by the utilization of a pointer having color tip (is red, green, or blue). The color tip is segment from back ground by converting RGB to HSI color model. Motion of color tip is identified by optical flow method. During formation of multiple gesture the unwanted lines are removed by optical flow method. The movement of tip is recoded by Motion History Image(MHI) method. After getting the complete gesture, then each character is extracted from hand written image by using the connected component and the features are extracted of the correspond character. The recognition is performed by minimum distance classifier method (Modified Hausdorff Distance). An audio format of each character is store in data-set so that during the classification, the corresponding audio of character will play.

Keywords: character recognition, Gesture, Optical flow, Motion history image, HSI color model, Connected Component, Modified Hausdorff Distance

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In computer vision research, out of various research area Human activity recognition is one of the important area. Human activities are divided in to four different levels such as gestures, actions, inter actions and groups activities [1]. Motions are rudimentary developments of a human body part, and are the atomic segments portraying the significant movement of a man. Movement of finger, extending an arm, and raising a leg are a few illustrations of signals.Activities are exercise that made various signals composed impermanent, for example strolling, waving and punching. Actions are human activities that have two or more persons for examples fighting between two persons, carry bags from someone else. Interaction exercise are human activities that include group of people for example discussion of an event in the committee.

The human activities which express meaningful body motion that involve physical movement of fingers, arms, head, face with the intent of meaningful expressing with environment. Those are constitute one interesting small subspace of human body motion.From environment gesture can be perceived as a compression technique for the information transmitted and reconstructed by receiver. In real life, gesture recognition has important applications, for example developing aids for the hearing impaired, recognizing sign language, lie detection, techniques for forensic identification.The gesture may be ambiguous and incomplete. For example to give sign stop, one may use gestures such as raised hand with palm facing forward or waving both hands over the head. Gestures can be static or dynamic. In static based gesture, a person assumes certain configuration, whereas in dynamic user associated with pre-stroke, stroke and post-stroke phases. Gestures are often language and culture specific. The gesture can be define in four ways (a) spatial information: where it occurs (b)temporal information: the path it takes (c) symbolic information: the sign it makes (d) affective information: its emotional quality. The same gesture can vary in shape and duration dynamically for the same person.

Table 1.1: Classification of gesture

Classification	Gesture details
hand and arm gesture	recognition of hand poses, sign languages
head and face gesture	nodding or shaking of head, direction of eye gaze, raising the eyebrows, opening the mouth to speak, winking, flaring the nostrils, looks of surprise, happiness, disgust, fear, anger, sadness
body gesture	involvement of full body gesture as tracking movement of two people interacting outdoors, analyzing movements of a dancer for generating matching music and graphics

In multidisciplinary research, the gesture is an ideal example. Human gesture typically constitute a physical movement of body parts such as face, hand, legs. Gesture can be classified as given in following list:

- *Gesticulation*: Extemporaneous movement of hands and arms, concomitant star with star speech. These Extemporaneous movements constitute around 90% of human motions. People gesticulate when they are on telephone, and even dumb people regularly make gesture when communicate with others.

- *Languagelike gestures*: Gesticulation interacted to a spoken utterance, replacing a particular spoken phrase.

- *Pantomimes*: Gesture is depicting object or actions, with or without accompanying speech.
- *Emblems*: The signs which are familiar like V for victory or other culture-specific gestures.
- *Sign languages*: Well-defined linguistic systems. These are the most semantic information and more systematic, so it is easier to model in virtual environment.

In this thesis, our focus is to identify Alphabetical character of English language through a color tip(Red, Green, Blue) gesture.

1.1 Related Work

Bobick *et al.* [1]has proposed a approach, which temporal template shows the motion and the pattern of movement. A temporal movement is constructed on template, where motion is define over frames, assuming either background remain static or motion of object can be from the camera-induced. Aggarwal *et al.* [2] have summarized various methodologies for the recognition of human activity, and discussed different approaches of the advantages and disadvantages of human activity recognition. All methodologies are classified such as (a) single-layered approaches and (b) hierarchical approaches. Human activities based on sequence of images are recognized by single-layered approaches. Human activities which are called sub-events are represented by hierarchical approaches. Mitra *et al.* [3] have described briefly the different tools and their use for gesture recognition. Shan *et al.* [4] has proposed for hand gesture recognition approach. The temporal motion of hand gesture is tracked, and then static image is presented using temporal template. Ishikawa *et al.* [5]have proposed a approach for the recognition of hand gesture by a data glove which measures the angle between the finger joint. The data glove has two sensors for the first and the second joint of each finger, in total it has ten angle sensors. Qureshi *et al.* [6] have proposed an algorithm for human hand gesture identification. The core work is the identification of finger which are active and those which are not. The peak or apex type pattern in fingers are identified which are regarded as joints of fingers. This joint detection is used to identify the active fingers. Sohn *et al.* [7]have proposed a scheme of 3D hand gesture recognition, in which from 3D depth camera the hand gesture video is obtained. Ahad *et al.* [8] have proposed an approach of motion segmented method, the optical flow is calculated based on motion history templates and sectioned into four directions: up, down, left, and right, and recognition of different gestures such as body stretching, waving arms, bending the chest has performed. The motion between a pixel in one frame and its correspondence pixel is calculated by optical flow method [9]. In [10] each gesture is defined to be an ordered sequence of state in spatio-temporal space. The 2D image positions of the center of the head and both hands are used as features; these are located by a color based tracking method. Lei *et al.* [11] have devised an accelerometer-based method for detecting the predefined one-stroke finger gestures, where data is collected using a MEMS 3D accelerometer, worn on the index finger. A compact wireless sensing mote integrated with the accelerometer, called magic ring, is developed to be worn on the finger for real data collection. A general definition on one-stroke gesture is given, and twelve kinds of one-stroke finger gestures are selected from human daily activities. Cemil *et al.* [12] have proposed recognition system, which is used a sensory glove called the cyber glove and a flock of birds to extracted the gesture feature. The bending angles of fingers at various positions is measured by the sensor of the glove. Out of the fifteen sensor of glove: three sensors for the thumb, two sensors for each of other four fingers, and four sensors between the fingers. The flock of birds motion tracker is mounted on the hand and wrist to track the position and orientation of he hands.

1.2 Motivation

It is observed that gesture is the natural form of communication. Controlling the home appliance, interaction with the computer is achieved using gesture. In the presences of sound noise gesture can works. For communicate with some electronics device dumb people use gesture. Usually hands and fingers are used to make numeral gesture. User require to wear a cumbersome device which connecting to the computer by lot of cable wires for making the gesture. From the literature, it is noted that sensors are used to make gestures and optical flow is a popular method to identify the motion of objects.

1.3 Objectives

In this thesis we have investigated to identify numerals through a gesture made by a fingertip or using pointer having a colored tip. In particular, the objectives of suggested scheme are narrowed to:

(a) Capture the gesture from a dynamic environment.

(b) Motion segmentation using optical flow mechanism.

(c) Formation of temporal template of numeral.

(d) Feature extraction.

(e) Classification and recognition of the extracted numeral.

The overall block structure is given in Figure **??** and phases are discussed below in nutshell.

(a) **Data acquisition :**

The presence of the five sensory organs of a human body helps to interact, learn and adapt with the challenging environment. The sight sensory organ helps in receiving visual information. This visual information can be captured and stored as an image by a camera. A single image is inadequate enough to represent a scene with motion information. Such scenes are recorded by capturing a sequence of images at regular intervals. Each image of the sequence is known as frame. When successive frames are projected with the progress of time, we call it as *video.* Projection of successive frames at a particular rate creates an illusion, which convey a sense of motion in the scene.

(b) **Segmentation :**

In computer vision, segmentation refers to the process of partitioning a digital image into multiple segments. It is the allocation of every pixel in an image, a label to which they correspond to a specific part. The goal of image segmentation is to partition the image into perceptually similar regions [13]. Segmentation is an extremely important operation in several applications of image processing and computer vision, since it represents the very first step of low-level processing of imagery [14]. Every segmentation algorithm addresses two problems, the criteria for a good partition and the method for achieving efficient partitioning [15].All images processing operations aim at good recognition of objects of interest, i.e., discovering suitable nearby components that can be recognized from different articles and from the background. The following step is to check every individual pixel, whether it belongs with an object of interest or not. Image segmentation produces a binary image, where one represents the object and zero represents the static background. Out of different approaches for segmentation thresholing, edge-based and region based method are most popular [16]. During thresholding pixels are allocated according to range of value in which pixel lies. Pixels are classified as edge or non edge depending upon filter output when edge-based segmentation applied to the image. Region-based segmentation operated by grouping pixels which are neighbors and have similar values.

In this thesis, segmentation of motion and color is carried out using thresholding mechanism. Motion is an integral part of video sequence. Segmentation of motion is a building block for robotics, visual surveillance, video indexing and many other applications. It provides a very rich set of information through which a wide variety of works are accomplished. Perceptual organization, 3D shape determination, scene understanding are to name a few. The three main issues in motion segmentation are data primitives or region of support [17]. The individual pixels, corners, lines, blocks or regions are data primitives. Motion model and motion representation are the second issue, which can be 2D optical flow, or 3D motion parameters involves motion estimation. And segment criteria is the third issue. The attributes of a motion segmentation algorithm is summarized in following list.

- $Feature-based$ or $Dense-based$: In this method objects are represented by limited number of points and it always compute a pixel-wise motion.
- *Multiple objects*: More than one object take participate in gesture formation.
- *Spatial continuity*: Accomplishment structural continuity.
- $Temporary\ stopping$: Deal with temporary stop of objects motion.
- *Robustness*: Deal with image contain noise (in case of feature based methods it is the position of the point to be affected by noise but not the data association).

In this thesis data primitives is individual pixel and motion is represented using 2D optical flow.

Color is one of the most distinctive clues in finding objects. Several color representations are currently used in color image processing. In RGB color space is represented by Red, Green, and Blue components in orthogonal Cartesian space. This is in agreement with the $tristimulus\ theory\ of\ color$ [18]

according to which human visualize system acquires color image by three band pass filter whose spectral tuned to wavelength of red, green and blue. The band pass filter (three different kinds of photoreceptors in the retina called cones) whose spectral responses are tuned to the wavelengths of red, green, and blue.

(c) **Temporal Template :**

It is a visual-based motion identification method is partial recognition method for recognition gestures without any incorporation of sensors on the human body. A view-specific representation of movement is constructed, where movement is defined as motion over time. The image sequence is converted into a static shape pattern [8].

(d) **Post processing :**

Some morphology operations [16] such as Dilation, Erosion, Thinning, Pruning are employed in order to have a invariant and stable representation.

- Dilation: an operation that grows or thickens object in an image.
- Erosion: an operation that shrinks or thins object in a binary image.
- Thinning: an operation in which binary valued image regions are reduced to lines that approximate the center skeletons of the regions [19] . It gives the skeleton representation of object that preserves the topology aiding synthesis and understanding.
- Pruning: an operation in which spur outliers are removed by setting pixel values to black. It is implemented by detecting end points and by removing them until idempotence [20].

(e) **Feature Extraction and Classification:**

feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative, non redundant, facilitating the subsequent learning and generalization steps, in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction. The feature is one or more measurements some quantifiable property of an object, and is computed such that it possesses some significant characteristic of the objects. The features are categorized in given list:

- **General features**: Application independent features such as color, texture, and shape. According to the abstraction level, they can be further divided into:
    - **Pixel-level features**: At each pixel features are calculated. e.g. color, shape. Local features: The images are sub-divided for feature extraction and edge detection.
    - **Local features**: Features calculated over the results of subdivision of the image band on image segmentation or edge detection.
    - **Global features**: The feature is calculated on the entire image or just sub-area of an image.
- **Domain-specific features**: Features such as human faces, fingertips are application dependent. For specific domain these features are synthesis of low-level feature.

All features are classified into two level, such as low-level features and high level features. The features which are extracted directly from the original images are low-level features, where as features extracted based on low-level features are high-level feature [21].

1.4 Thesis Organization

The overall thesis is organized into five chapters including the introduction.

Chapter 2 presents the formation of alphabetical character(English alphabet) using pointer having color tip(red, green, blue) gesture. Motion of tip is obtained using 2D motion vector. Temporal template of the characters are formed and morphological operation is performed. Motion of tip is captured through a video, but the other moving parts are removed to extract the motion of the tip.

Chapter 3 presents the formation of multiple character and also show the removal of unwanted lines in single character and multiple characters. However the extraction of the desired tip is extracted.

Chapter 4 deals with the feature extraction from the final template and its classification is studied.

Chapter 5 presents the concluding remarks with scope for future research work.

Chapter 2

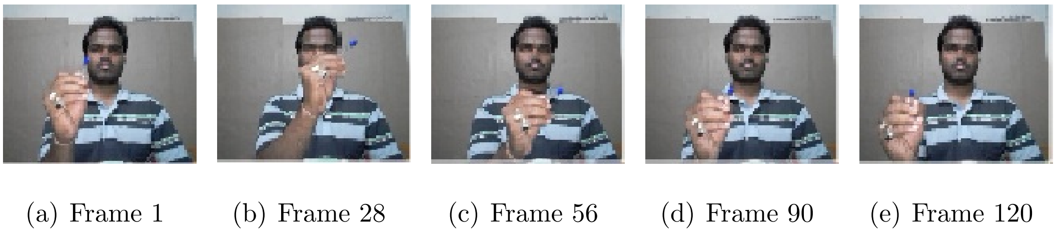
Formation of Characters Gesture using Color Tip

In this chapter, we exploit all the phases/steps required for alphabetical character formation using color tip(red, green, blue) gesture. The steps in order are given below.

- Video data acquisition
- Color Segmentation
- Motion segmentation using optical flow method
- Motion history image formation

2.1 Video Acquisition

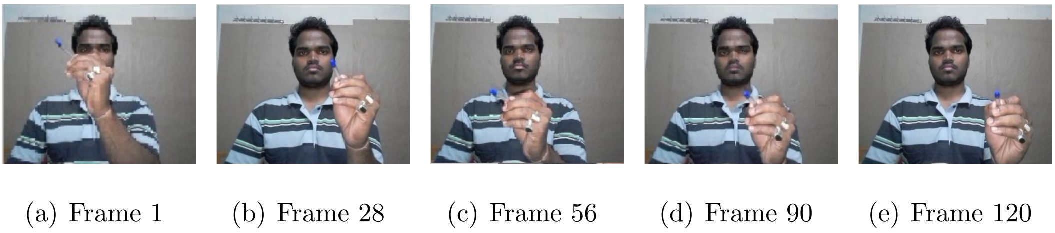
Video are taken as input data in our system. In the video the motion of tip of pointer whose color either red, green or blue is captured using a mobile camera having resolution 5M pixel. The motion is in such a way that it makes the gesture of a particular alphabet as shown in Figure 2.1, which symbolize alphabet 'Z' in terms of frames.



(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.1: Frames of color tip gesture of original video.

The input video captured by camera is stored in inverted form. Prior to motion segmentation the captured video is preprocessed to inverted form using Algorithm 1 and the result is shown in Figure 2.2.



(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.2: Frames of color tip gesture of inverted video.

Algorithm 1 Correction of Captured video

```
Input : Captured Video $O$
Output : inverted imaged video $U$
$x \leftarrow number\ of\ column\ of\ each\ frame$
$d \leftarrow x$
for $l \leftarrow 1\ to\ number\ of\ frame$ do
  for $m \leftarrow 1\ to\ height\ of\ frame$ do
    for $n \leftarrow 1\ to\ width\ of\ frame$ do
      $U(m, n, :, l) \leftarrow O(m, x, :, l)$
      $x \leftarrow x - 1$
    end for
    $x \leftarrow d$
  end for
end for
```

Here our objective is to achieve motion segmentation using both color and brightness information.

2.2 Color Segmentation

Color is perceived as a combination of three color stimuli: red, green, and blue, which forms a color space. RGB colors are called primary colors and are additive. Figure 2.3 shows the RGB color model. By varying their combinations, other colors can be obtained. Color is characterized by three quantities.

- **Hue**: It is an attribute that defines pure color. It is associated with the dominant wavelength in a mixture of light waves. It represents the dominant color perceived by observer, i.e whenever we call an object red, green or blue, we refer to its hue.

- **Saturation**: It gives a measure of degree to which a pure color is diluted with white light. It is inversely proportional to the amount of white light added.
- **Brightness**: It is the achromatic notion of intensity and is one of the key factor to describe color sensation.

A color model is a specification of a co-ordinate system within which each color is represented by a single point.The RGB space does not give itself the larger amount forms which request the view of shading regarding the human visual framework. In term of hue, saturation, and intensity presentation is the best way [16,22]. HSI color space is an example of such representation.

2.2.1 HSI (Hue, Saturation, Intensity) color model

The HSI color space, decouples the intensity component from the color carrying information (hue and saturation) in a color image [16]. An RGB color image is composed of three monochrome intensity images, in which intensity is extracted from RGB image. The color cube along the black (0,0,0) in Figure 2.3, with the white vertex (1,1,1), directly above black vertex, in Figure 2.3. The intensity is along the line joining these two vertices. The intensity component of any color point determine by pass the plane in intensity axis gives the intensity value. Saturation of a color increases as a function of distance from intensity axis.

On the intensity axis where the saturation of points are zero, all the points are shading gray value along the axis. To determine from given RGB points, consider Figure 2.3(b), which defined by three points, (Black, white, and cyan). While there are black and white shades that shows that the intensity axis contained in that plan. All points contained in the plane segment defined by the intensity axis and the boundaries of the cube have same hue. This is because the colors inside a color triangle are various combinations or mixtures of the three vertex colors. If two of those vertices are black and white, and third is a color point, all points on the triangle must have the same hue because black and white components do

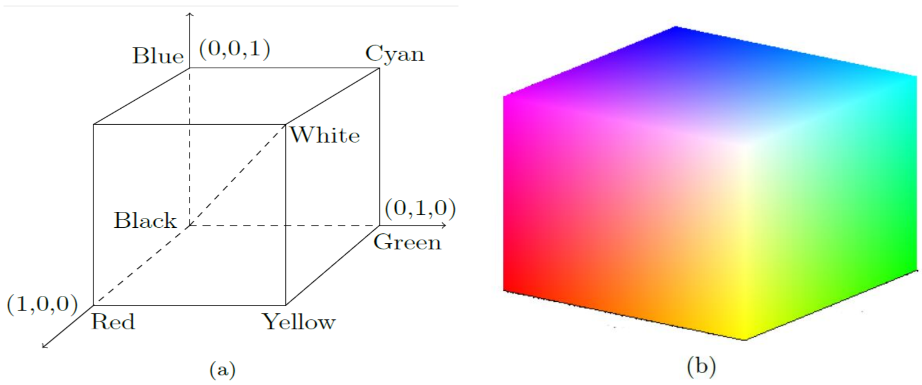


Figure 2.3: (a) The RGB color cube which shows colors of light at vertices. At origin, the main diagonal has gray value from black, at origin to white at point (1,1,1) (b) The RGB color cube.

not contribute to changes in hue. By rotating the shaded plane about the vertical intensity axis, different hue value are obtained.

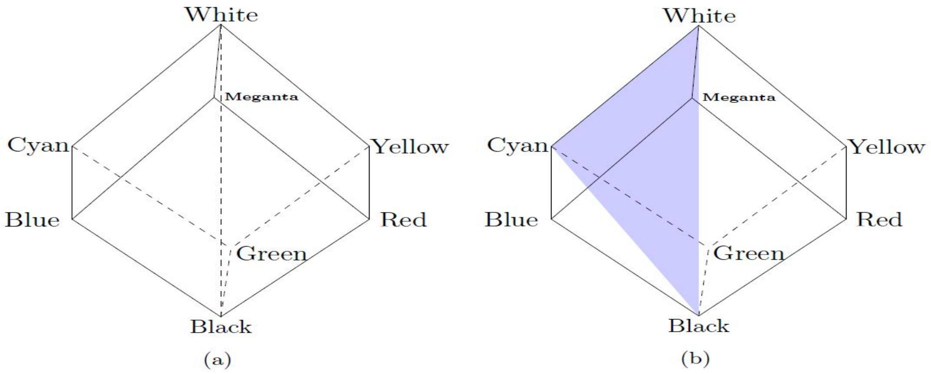


Figure 2.4: Relationship between RGB and HSI color model.

The HSI space consists of a vertical intensity axis and the locus of color points that lie in a plane perpendicular to this axis. As the plane moves up and down the intensity axis, the boundaries defined by the intersection of the plane with the faces of the cube have either a triangular or hexagonal shape. This can be visualized much more by looking at the cube down its gray scale axis, as shown in Figure2.4(a).

In this plane angle between secondary colors is 120 °, because on the plane primary colors is separated by 120 ° and the secondary colors are separated by 60 ° from the primaries. The hexagonal shape and arbitrary color points as shown in Figure 2.4(b). From some reference points the hue of point is determined. At angle of 0 ° from intensity axis represent red color and designates 0 hue, and increases counter clockwise subsequently. From the origin to the point, the saturation is the length of vector. The interaction of object plane with the vertical intensity axis is defined the origin. Vertical intensity axis, length of the vector to a color point and the angle this vector makes with the red axis are the main components of the HSI color space. Given an image in RGB color format is converted into HSI model as,

$$H = \begin{cases} \theta & if \quad B \leq G \\ 360 - \theta & if \quad B > G \end{cases} \tag{2.1}$$

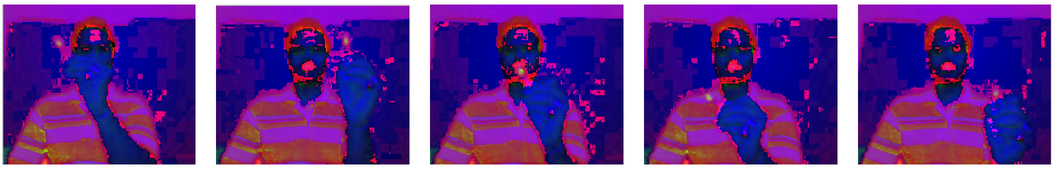
where

$$\theta = \cos^{-1}\left\{ \frac{0.5 \times [(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{(R+G+B)}[minimum(R,G,B)] \tag{2.2}$$

$$I = \frac{1}{3}(R+G+B) \tag{2.3}$$

RGB color model is converted into HSI color space using equations (2.1), (2.2), and (2.3) as shown in Figure 2.5. The segmentation of color is carried out using Algorithm 2, and the result is shown in Figure 2.6.

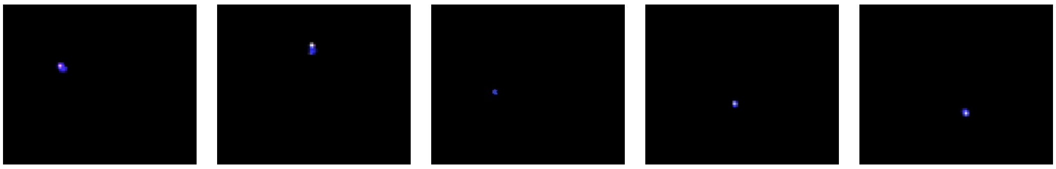


(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.5: Frames of Video in HSI color model.

Algorithm 2 Color Segmentation

```
Input : processed video $U$ and video $L$ in HSI color model
Output : Decomposed video $W$ in RGB color model
for $k \leftarrow 1$ to number of frame do
    for $l \leftarrow 1$ to height of frame do
        for $m \leftarrow 1$ to width of frame do
            if $L(l,h,1,k) > \tau_1$ and $L(l,h,2,k) > \tau_2$ and $L(l,h,3,k) > \tau_3$ then
                $W(l,h,:,k) \leftarrow U(l,h,:,k)$
            else
                $W(l,h,:,k) \leftarrow 0$
            end if
        end for
    end for
end for
```



(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.6: Frames of video after color segmentation.

2.3 Motion Segmentation

In motion segmentatio a video is decompose to moving objects and background [23].In a three-dimensional co-ordinate system, when an object move, then it is projected on an image plane and each points produce a two-dimensional path. Usually 2D motion field contain instantaneous direction of points velocity and 2D velocity at all points. Optical flow method is employed to estimate an approximation of the motion field from a set of images varying with respect to time. It is a 2D vector which gives the displacement of each pixel with respect to its previous frame. Optical flow is Optical flow is the distribution of velocities for each pixel movement of brightness pattern [9]. It arises due to relative motion of each pixel. If the camera, or an object, moves within the scene, this motion results in a time dependent displacement of the gray values in the image sequence. The resulting two-dimensional apparent motion field in the image domain is called the optical flow field. There are various methods to compute optical flow given in literature [24–28]. It is a dense field of displacement vectors which defines the translation of each pixel in a region.

2.3.1 Computation of optical flow

Three popular methods to compute optical flow are Horn and Schunck method [9], Lucas and Kanade Window method (LKW) [29], and Least Square Fit Method (LKF) [30] are implemented towards the simulation of the proposed work discussed in detail below in sequence.

(a) **Horn and Schunck method**: This method based on two assumptions, one is brightness consistency and velocity smoothness. For better understanding we describe both below in nutshell.

- **Brightness constancy**: The brightness is constant over the time at any pixel on a image. Let F($x1$, $y1$, $t1$) is brightness at images point ($x1$, $y1$) at time $t1$ and images move dx in x-direction, dy in y-during interval dt, then

$$F(x1 + dx, y1 + dy, t1 + dt) = F(x1, y1, t1) \tag{2.4}$$

Using Taylor series expansion and neglecting higher order terms yields

$$(\partial F1/\partial x).dx + (\partial F1/\partial y).dy + (\partial F1/\partial t).dt = 0 \tag{2.5}$$

For simple notation, let

$$(\partial F1/\partial x) = f_x,\ (\partial F1/\partial y) = f_y,\ (\partial F1/\partial t) = f_t$$

Using this notation and dividing equation (2.5) with dt, we get

$$f_x1.dx/dt + f_y1.dy/dt + f_t1 = 0 \tag{2.6}$$

Let

$$dx/dt = u1,\ dy/dt = v1$$

So the equation (2.5) became

$$f_x1.u1 + f_y1.v1 + f_t1 = 0 \tag{2.7}$$

where *u1* and *v1* is the velocity components of each pixel in the *x1* and *y1* direction and (f_x1, f_y1) is the rate of change of brightness with respect to time at a point in the image.

Figure 2.7 shows equation (2.7) is a straight line with *u1* as x-axis and *v1* as y-axis. Optical flow of point P can be anywhere on the straight line. Point P has two types of flow; parallel flow, which is along the straight line and normal flow, perpendicular to the straight line. Normal flow is not changing, the distance D remains constant, however parallel flow is changing which need to be computed. In equation (2.7), f_x1, f_y1, and f_t1 are known and unknown variables are $u1$ and $v1$, so it is an under constrained equation. To get the unknown variables $u1$ and $v1$ at least two equations are required, i.e. an additional constraint is required.

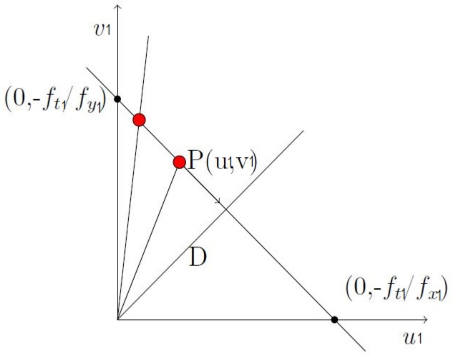


Figure 2.7: Interpretation of optical flow equation.

- **Velocity smoothness**: Neighbour pixels in the image plane move in a similar manner and in other word this constraint is to minimize the square of the magnitude of the gradient of the optical flow velocity.

$$(\partial u1/\partial x)^2 + (\partial u1/\partial y)^2 \; and \; (\partial v1/\partial x)^2 + (\partial v1/\partial y)^2 \tag{2.8}$$

Ideally equation (2.7) has to be zero, but in practical scenario it is not, also there is deviation from smoothness in the velocity flow given in equation (2.8), so the total error to be minimized is:

$$\int\int (f_x1.u1 + f_y1.v1 + f_t1)^2 + ((\partial u1/\partial x)^2 + (\partial u1/\partial y)^2 + (\partial v1/\partial x)^2 + (\partial v1/\partial y)^2)dxdy \tag{2.9}$$

solving the equation (2.9), we get

$$(f_x1.u + f_y1.v + f_t1).f_x1 + \alpha(u1 - u1_{avg}) = 0 \tag{2.10}$$

$$(f_x1.u1 + f_y1.v1 + f_t1).f_y1 + \alpha(v1 - v1_{avg}) = 0 \tag{2.11}$$

where α is a smoothness constraint. In order to compute $u1_{avg}$ and $v1_{avg}$ for each pixel find its 4-neighborhood, add all the pixel value and divide the sum by 4.

(b) **Lucas and Kanade Window method (LKW)**: Equation (2.7) can be written as

$$f_x 1.u1 + f_y 1.v1 = -f_t 1 \tag{2.12}$$

Lucas and Kanade has assumed that motion is smooth locally i.e. motion vectors in a given region do not change but merely shift from one position to another. For a given pixel we look around its $n \times n$ neighbor with n >1 and assume optical flow on these pixel is same. For example, consider a 3×3 window, the set of equations are,

$$\begin{aligned} f_{x_1}.u &+ f_{y_1}.v = -f_{t_1} \qquad (2.13)\\ f_{x_2}.u &+ f_{y_2}.v = -f_{t_2} \qquad (2.14)\\ &\vdots \\ f_{x_2}.u &+ f_{y_2}.v = -f_{t_9} \qquad (2.15) \end{aligned}$$

This system of equations can be written as:

$$\begin{aligned} f_{x_1}.u &+ f_{y_1}.v = -f_{t_1} \qquad (2.16)\\ f_{x_2}.u &+ f_{y_2}.v = -f_{t_2} \qquad (2.17)\\ &\vdots \\ f_{x_2}.u &+ f_{y_2}.v = -f_{t_9} \qquad (2.18) \end{aligned}$$

This system of equations can be written as:

$$\begin{bmatrix} f_{x_1} & f_{y_1} \\ \vdots & \vdots \\ f_{x_9} & f_{y_9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t_1} \\ \vdots \\ -f_{t_9} \end{bmatrix} \tag{2.19}$$

$$AU = f_+ \tag{2.20}$$

where $A = \begin{bmatrix} f_{x_1} & f_{y_1} \\ \vdots & \vdots \\ f_{x_9} & f_{y_9} \end{bmatrix}$, $U = \begin{bmatrix} u \\ v \end{bmatrix}$, $f_+ = \begin{bmatrix} -f_{t_1} \\ \vdots \\ -f_{t_9} \end{bmatrix}$

Now, vector U can be computed using the pseudo inverse method as follows:

$$A'AU = A'f_+$$

$$U = (A'A)^{-1}A'f_+ \tag{2.21}$$

(c) **Least Square Fit Method (LSF)**: Ideally equation (2.19) should be zero but it is not happening i.e. error are there because we are estimating u and v. In some equation it is positive and in some it is negative so we square the error and sum it.

$$minimize \sum_{i=1}^{n^2} (f_{x_i}u + f_{y_i}v + f_{t_i})^2 \tag{2.22}$$

Differentiating equation (2.22) with respect to u and v separately, final equation we get

$$\sum (f_{x_i}u + f_{y_i}v + f_{t_i})f_{x_i} = 0 \tag{2.23}$$

$$\sum (f_{x_i}u + f_{y_i}v + f_{t_i})f_{y_i} = 0 \tag{2.24}$$

This system of equations can be written as:

$$\begin{bmatrix} \sum (f_{x_i})^2 & \sum f_{x_i}f_{y_i} \\ \sum f_{x_i}f_{y_i} & \sum (f_{y_i})^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum f_{x_i}f_{t_i} \\ -\sum f_{y_i}f_{t_i} \end{bmatrix} \tag{2.25}$$

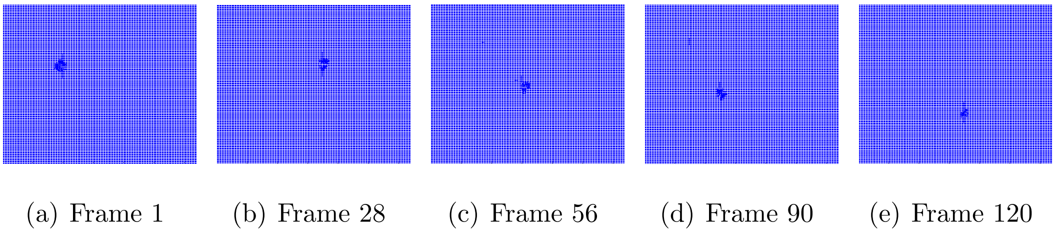
$$BU = f \tag{2.26}$$

where, $B = \begin{bmatrix} \sum (f_{x_i})^2 & \sum f_{x_i}f_{y_i} \\ \sum f_{x_i}f_{y_i} & \sum (f_{y_i})^2 \end{bmatrix}$, $U = \begin{bmatrix} u \\ v \end{bmatrix}$, $f = \begin{bmatrix} -\sum f_{x_i}f_{t_i} \\ -\sum f_{y_i}f_{t_i} \end{bmatrix}$

Equation (2.26) gives the optical flow of each pixel.

Horn and Schunck method gives the global information and smooth flow, whereas non-iterative Lucas and Kanade method gives the local information. The latter one does not yield a very high density of flow vectors. Least square fit is an extension of

Lucas and Kanade method which minimizes the error produced by LKW method. Due to smooth and higher density of flow vector, the method of Horn and Schunck is found to perform well. The optical flow estimated by Horn and Schunck method of the preprocessed frames of the video is given in Figure 2.8 and suitable thresholding is performed to get the region of interest.



(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.8: Optical flow between frame (1, 2), frame (28, 29), frame (56, 57), frame (90, 91), and frame (119, 120).

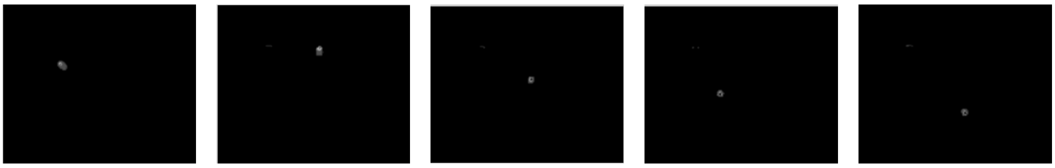
2.3.2 Thresholding

For each pixel along x and y directions, $u1$ and $v1$ are the optical flow respectively. The magnitude of optical flow for each pixel is given by

$$M = \sqrt{u1^2 + v1^2} \tag{2.27}$$

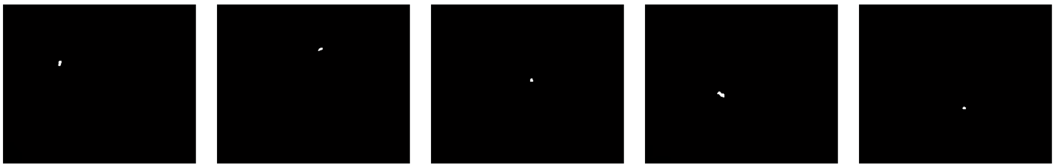
For computing the motion of each pixel the magnitude M has been assigned as pixel intensity values, thus resulting in a sequence of gray scale images as shown in Figure 2.9.

Higher the value of $u1$ and $v1$, the higher is the magnitude M of motion and hence more prominent is the pixel motion in the corresponding gray scale image. Further the region of interest is segmented using Algorithm 3 and shown in Figure 2.10.



(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.9: Gray scale form of finger gesture.



(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.10: Frames showing prominent motion after thresholding.

Algorithm 3 Representaion of image intensity and thresholding

```
Input : Computed $(u, v)$ as pixel velocity components in x and y direction respectively
Output : Prominent motion after thresholding
for $j \leftarrow 1$ to height of frame do
    for $k \leftarrow 1$ to width of frame do
        $z_u(j,k) \leftarrow u(j,k) * u(j,k)$
        $z_v(j,k) \leftarrow v(j,k) * v(j,k)$
        $mag(j,k) \leftarrow \sqrt{(z_u(j,k) + z_v(j,k)}$
    end for
end for
$Q1 \leftarrow maximum\ (Mag)$
$Q2 \leftarrow minimum\ (Mag)$
for $j \leftarrow 1$ to height of frame do
    for $k \leftarrow 1$ to width of frame do
        $MAG(j,k) \leftarrow\ mag(j,k)/(Q1 - Q2)$
        if $MAG(j,K)\ <= \tau 1$ then
            $MAG(j,k) \leftarrow 0$
        end if
    end for
end for
```

Morphological operation is performed to evacuate the undesirable movement which is still left after movement decomposition. The following step is to change over the image arrangement fit as a static shape pattern and it is accomplished utilizing movement history picture [8].

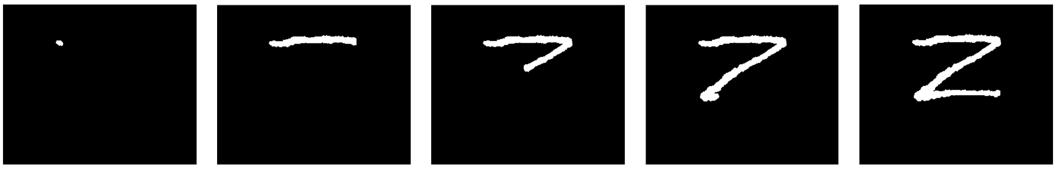
2.4 Motion History Image (MHI)

The motion history image (MHI) approach is a view-based temporal template method which is simple but robust in representing movements and is widely employed by various research groups for action recognition, motion analysis and other related applications [8]. MHI records the temporal history of motion and also describes *how* the object is moving. Approaches based on template matching first convert an image sequence into a static shape pattern and then compare it to the pre-stored action prototypes during recognition. During formation of MHI dominant motion information is preserved and silhouette sequence is condensed into gray scal image. It can represent a motion sequence of object in compact manner. Noise like holes, shadows, and missing parts are not sensitive for MHI template. It records the temporal changes at each pixel location in image, which then decays over time. By using the intensity, the MHI express the motion flow or sequence of every pixel in temporal manner. MHI keeps a scope of times encoded in a single frame, which is a most advantage of MHI. It compasses the time size of human gesture. The motion history recognizes general patterns of movement; thus, it can be implemented with cheap cameras and lower powered CPUs [31]. This method does not need trajectory analysis [32]. The Motion History Image(MHI) is computed upon updated function ψ(x1,y1,t1), which represents the optical flow.

$$MH(x1, y1, t1) = \begin{cases} \tau & if \quad \psi(x1, y1, t1) = 1 \\ \max(0, MH(x1, y1, t1-1) - \delta) & otherwise \end{cases}$$

where, $(x1, y1)$ shows position of time and $t1$ shows time, τ is the temporal extent of the movement for example number of frames, and δ is the decay operator. The result of this computation is a scalar-valued image, where more recently moving pixels are brighter and vice-versa [1, 33]. The recursive definition implies that no history of the previous images or their motion fields needs to be stored nor manipulated, which makes the computation fast and space efficient. Figure 2.4

shows the formation of motion history image of alphabet 'Z'.

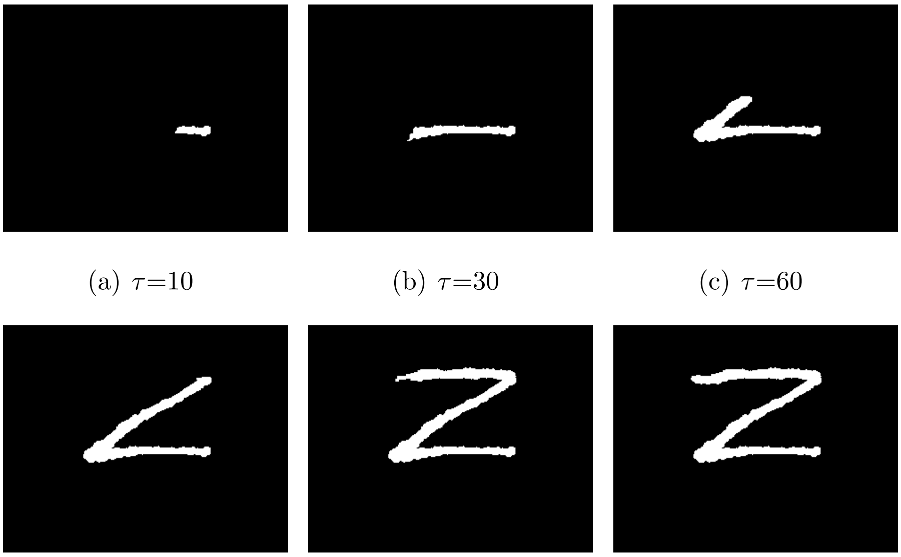


(a) Frame 1 (b) Frame 28 (c) Frame 56 (d) Frame 90 (e) Frame 120

Figure 2.11: Frames showing the formation of MHI.

2.4.1 Effect of τ and δ on MHI

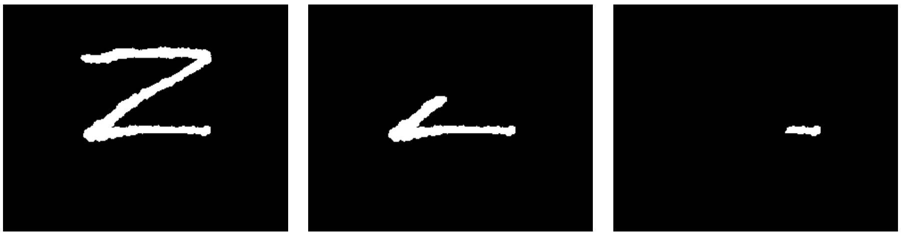
Different τ values produce for different MHI. If the nuber of frames more than the τ, then the prior information of gesture is lost in its MHI. Figure 2.12 shows the dependence on τ in producing the MHI. For example, when τ =20 for a gesture having 120 frames, there is a loss of motion information after 20 frames where the value of decay parameter δ is 1. On the other hand, if the temporal duration value is set at very high value compared to the number of frames, for example 500 then it is less significant with the changes of pixel values in MHI template. Figure 2.13 shows the calculating the MHI image and dependence of decay parameter δ. When earlier there was motion and then no change of motion in pixel, the pixel value is reduce by δ. However, having different δ values may provide slightly different information; hence the value can be chosen empirically. It is evident from Figure 2.13 that higher values for δ remove earlier trail of motion sequence. This information is important based on the demand and action, we can modulate the value of δ and τ. Here we take τ and δ is taken as 500 and 1 respectively for the formation of motion history image.



(a) τ=10 (b) τ=30 (c) τ=60

(d) τ=90 (e) τ=120 (f) τ=500

Figure 2.12: Effect of τ in calculating MHI template where δ=1.

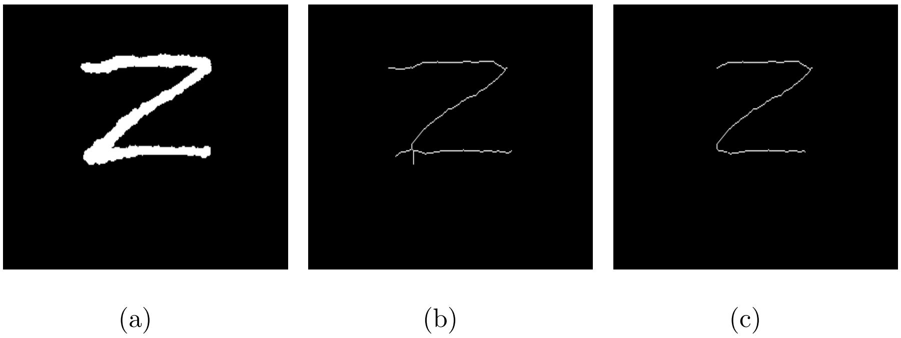


(a) δ=1 (b) δ=4 (c) δ=8

Figure 2.13: Effect of δ in calculating MHI template.

2.4.2 Post processing

Thickness of the gesture vary among gestures, so to bring consistency thinning [34] is execute. Thinning is a morphological operation in which binary valued image regions are reduced to lines that approximate the center skeletons of the regions [19]. It outputs the thinnest representation of gesture that protect the topology supporting synthesis as indicated in Figure 2.14(b). Spurs [19] remove in image by setting the pixel worth to dark utilizing the pruning operation which is shown in Figure 2.14(c).



(a) (b) (c)

Figure 2.14: Final post processing on MHI (Horn and Schunck method).

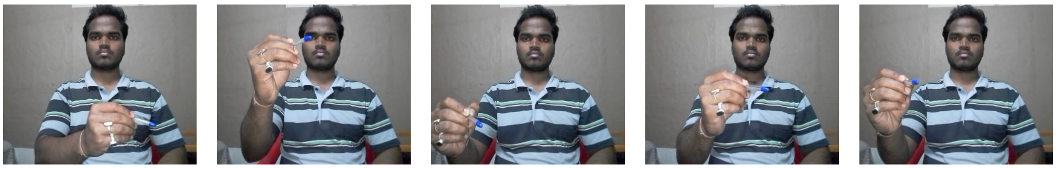
2.5 Summary

In this chapter the formation of Character(alphabet) using color tip gesture is presented where acquisition of video is done using mobile camera having resolution 5M Pixel, so acquisition device is not a limitation.The color tip is segmented by converting the videom from RGB model to HSI model. In the video motion of color tip(red,green,blue) is captured which is obtained using the Horn and Schunck method presented in this chapter. Temporal history of motion is recorded using motion history image and finally post processing is done to get a better thinned image.

Chapter 3

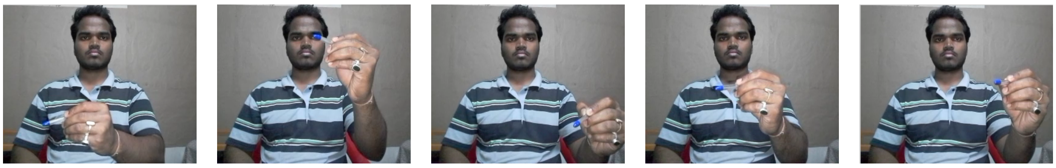
Remove Unwanted line from Gesture

In this chapter, we have proposed a scheme, in which removing unwanted line in gesture formed during the formation of alphabetical character. Input video is captured using a mobile camera with 5M pixel resolution. In the previous chapter, we have performed segmentation using both color and brightness information. Here our objective is to achieve multiple gesture by removing the unwanted lines. Figure 3.1 shows the frames of a video in which gesture of alphabet 'A' is performed. The input video is preprocessed to convert it into its true form using Algorithm 1 and the result is shown in Figure 3.2.



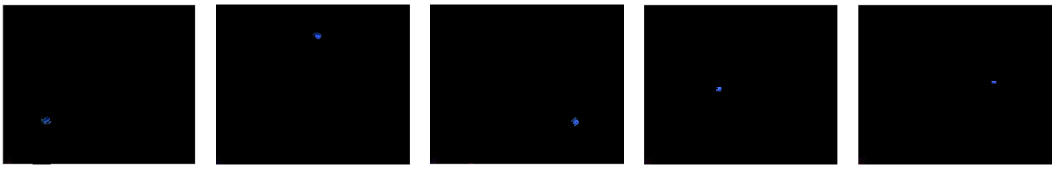
(a) Frame 1 (b) Frame 110 (c) Frame 230 (d) Frame 305 (e) Frame 410

Figure 3.1: Frames of Input video.



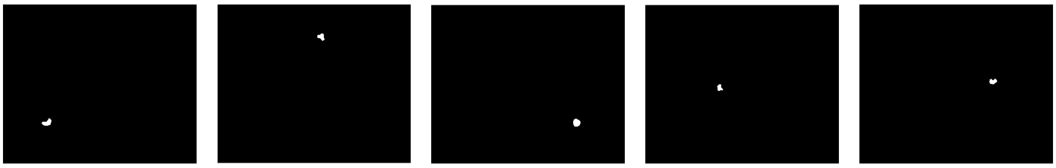
(a) Frame 1 (b) Frame 110 (c) Frame 230 (d) Frame 305 (e) Frame 410

Figure 3.2: Frames of Inverted video.



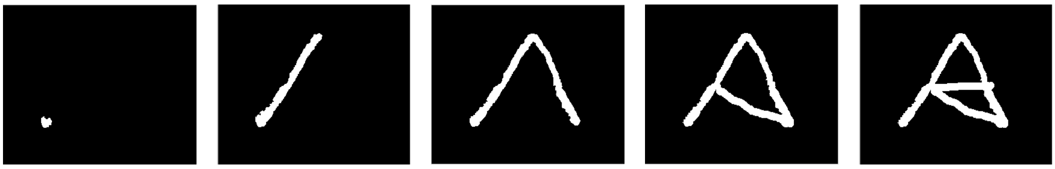
(a) Frame 1 (b) Frame 110 (c) Frame 230 (d) Frame 305 (e) Frame 410

Figure 3.3: Frames of color segmented video.



(a) Frame 1 (b) Frame 110 (c) Frame 230 (d) Frame 305 (e) Frame 410

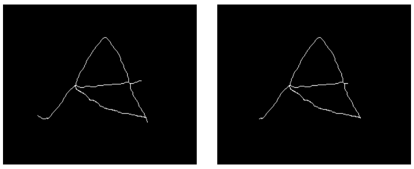
Figure 3.4: Frames of Inverted video.



(a) Frame 1 (b) Frame 110 (c) Frame 230 (d) Frame 305 (e) Frame 410

Figure 3.5: Frames of MHI video.

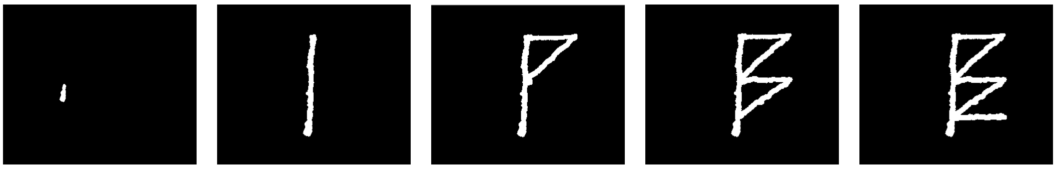
After processing thinning operation on MHI template, then binary valued image regions are reduced to line. Figure 3.6(a) show thinning operation and Figure 3.6(b) show pruning operation. After pruning operation unwanted spurs are removed.



(a) thinning (b) pruning

Figure 3.6: Morphological Operations

Likewise English alphabet 'E' is formed, in which unwanted lines appear during gesture formation. shown in Figure 3.7.



(a) Frame 1 (b) Frame 110 (c) Frame 230 (d) Frame 305 (e) Frame 410

Figure 3.7: Frames of alphabet 'E'showing unwanted lines video.

During formation of multiple gesture, some uninterested lines appear. Removing of uninterested lines will discuss for both single character and multiple characters in detail.

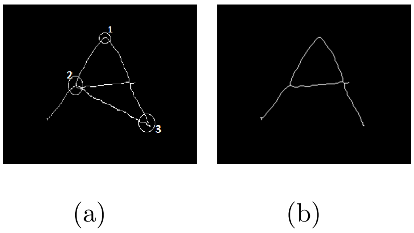
Previous chapter we performed motion segmentation by using optical flow method. Optical flow method is a 2D vector which gives the displacement of each pixel with respect to its previous frame. In other words optical flow is the distribution of apparent velocities of movement of brightness pattern [9]. Here we proposed algorithm, which remove uninterested line formed during formation of alphabetical gesture.

Algorithm 4 Remove unwanted lines

```
Input : Computed $(u, v)$ as pixel velocity components in x and y direction respectively
Output : Prominent motion after thresholding
$c \leftarrow 1$
$start \leftarrow 0$
$end \leftarrow 0$
for $k \leftarrow 2$ $to\ total\ number\ of\ frame$ do
    for $i \leftarrow 1$ $to\ height\ of\ frame$ do
        for $j \leftarrow 1$ $to\ width\ of\ frame$ do
            $z_u(i, j) \leftarrow u(i, j) * u(i, j)$
            $z_v(i, j) \leftarrow v(i, j) * v(i, j)$
            $mag(i, j) \leftarrow \sqrt{(z_u(i, j) + z_v(i, j)}$
        end for
    end for
    $magavg(c) \leftarrow average\ (mag)\ for\ each\ 30\ framea\ (frames/sec)$
    if $magavg(c) <= \tau_1$ then
        $flage = 1$
    end if
    if $flag == 1$ $and\ magavg(c) >= \tau_1$ then
        $Start = k$
        $flag = 0$
        if $start > end$ then
            $end = start$
        end if
    end if
    $q1 \leftarrow maximum\ (mag)$
    $q2 \leftarrow minimum\ (mag)$
    for $i \leftarrow 1$ $to\ height\ of\ frame$ do
        for $j \leftarrow 1$ $to\ width\ of\ frame$ do
            $MAG(i, j) \leftarrow mag(i, j)/(q1 - q2)$
            if $MAG(i, j) <= \tau$ then
                $MAG(i, j) \leftarrow 0$
            end if
        end for
    end for
    if $k == start$ then
        for $i \leftarrow 1$ $to\ height\ of\ frame$ do
            for $j \leftarrow 1$ $to\ width\ of\ frame$ do
                $MAG(i, j) \leftarrow mag(i, j)/(q1 - q2)$
                $MAG(i, j) \leftarrow 0$
            end for
        end for
        if $k == end$ then
            $continue$
        end if
    end if
end for
```

3.1 For single gesture

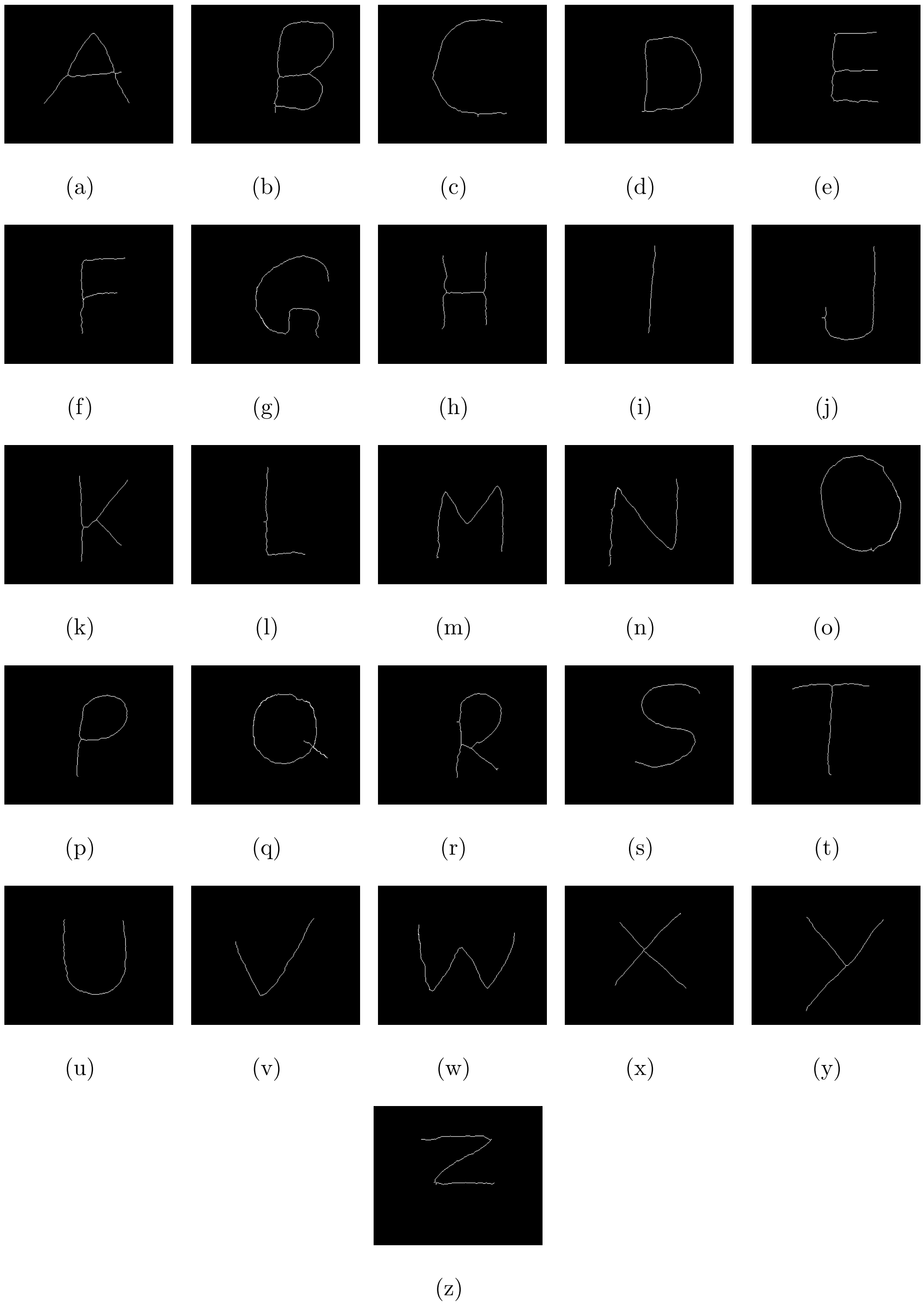
In previous section we performed that, during the formation of alphabetical gesture unwanted lines formed, which is not required during the extraction of features. During the formation of alphabetical gesture shown in Figure 3.8, optical value of frame will be minimum at points 1,2,3 shown in Figure 3.1 . But line from point 2 to 3 is not required. Here our objective is to remove the line between points 2 and 3. The processed color segmented video is formed using the Algorithm 4 and the result is then processed in MHI. Then the result is shown in Figure 3.1



(a) (b)

Figure 3.8: (a) Showing minimum optical value. (b) True form of gesture

Likewise, the other English alphabet is formed shown in Figure 3.1.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

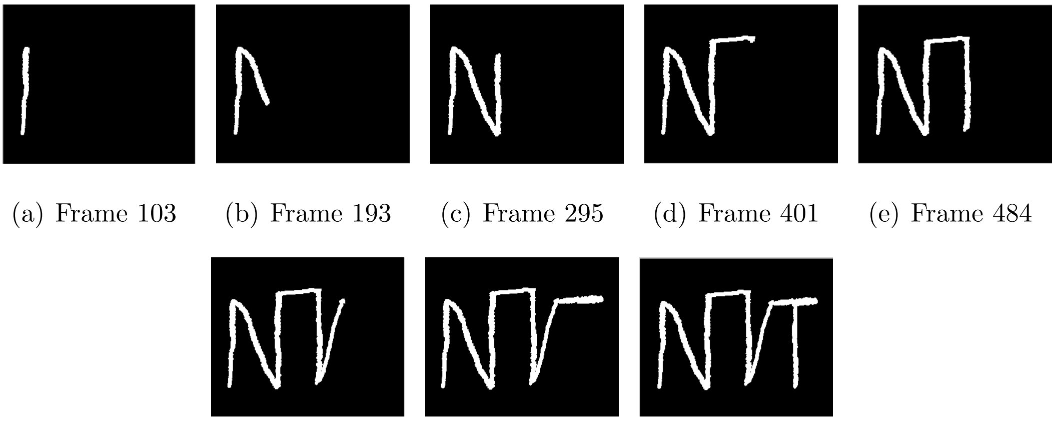
(u) (v) (w) (x) (y)

(z)

Figure 3.9: From (a)-(z) frames showing English alphabets A to Z respectively

3.2 For multiple gesture

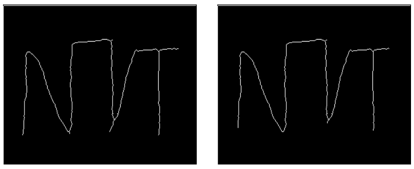
In previous section we performed that, uninterested lines are remove by using the Algorithm 4 in single gesture alphabets. During the formation of multiple gesture, there are also some uninterested line appeared in MHI templet. In Figure 3.2 showing some unwanted lines during formation of multiple gesture. Figure 3.2 shows the frames of MHI video in which gesture of multiple alphabets 'NIT' is performed. The Unwanted lines appeared on gesture can be remove by using Algorithm 4 and the result is shown in Figure 3.2.



(a) Frame 103 (b) Frame 193 (c) Frame 295 (d) Frame 401 (e) Frame 484

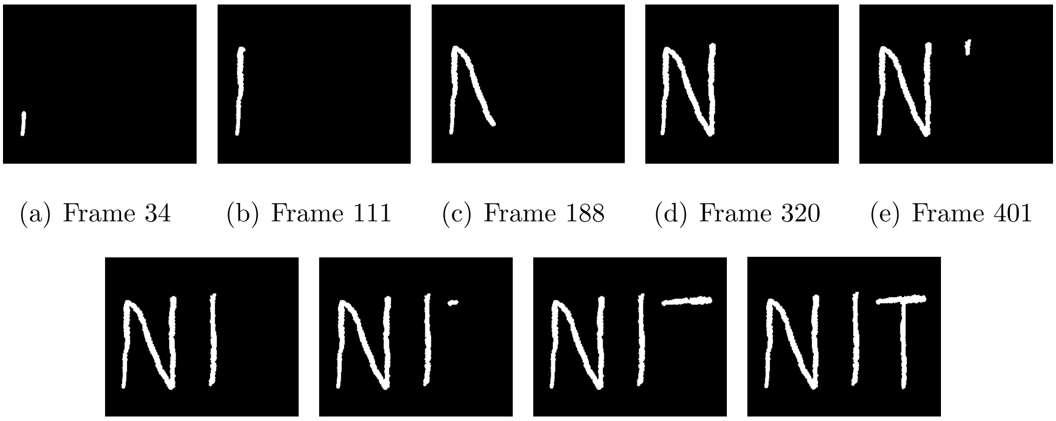
(f) Frame 597 (g) Frame 672 (h) Frame 806

Figure 3.10: Frame of MHI vedio, showing uninterested lines



(a) Thinning (b) Prunning

Figure 3.11: Morphological Operations

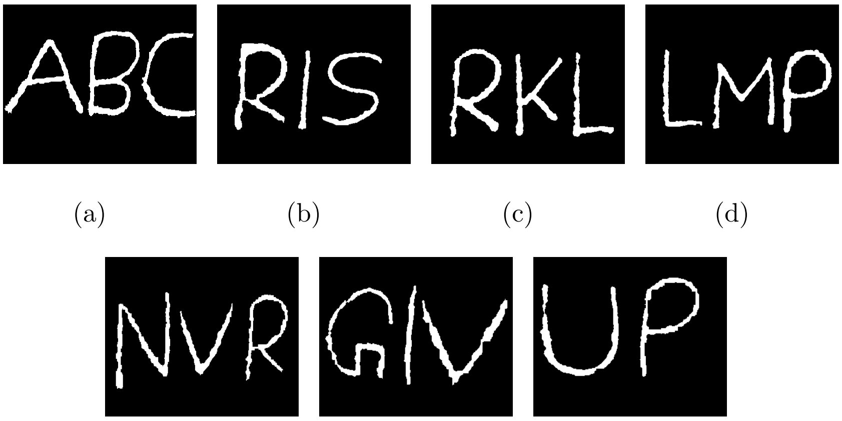


(a) Frame 34 (b) Frame 111 (c) Frame 188 (d) Frame 320 (e) Frame 401

(f) Frame 484 (g) Frame 606 (h) Frame 672 (i) Frame 806

Figure 3.12: Frame of MHI vedio, after removing uninterested lines

Likewise, the other multiple alphabet are formed as shown in Figure 3.13.



(a) (b) (c) (d)

(e) (f) (g)

Figure 3.13: Frame of MHI vedio, (a)-(g) Frames showing multiple gesture

3.3 Summary

In this chapter the formation of Character(alphabet) using color tip gesture is presented where unwanted lines appeared in MHI template. Here we have performed how the unwanted lines appeared both in single character and multiple character. And by using the proposed Algorithm 4, the unwanted lines are removed. English alphabetical characters A-Z are formed by using the proposed algorithm. Similarly in multiple gesture uninterested lines appeared and these lines can be removed by Algorithm 4. Then by using the proposed algorithm we performed different words.

Chapter 4

Feature Extraction and Classification

Features are the inherent property of data. Transforming the input data into the set of features is called feature extraction. Feature extraction involves simplifying the amount of information required to describe an input image. In this chapter objective is to extract each character from MHI template and finding the feature vector of the corresponding character. Each character extracted from hand written gesture by connected component.

4.1 Connected Component

Connected component labeling is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic. Connected component labeling is used in compute vision to detect connected regions in binary digital images, although color images and data with higher dimensionality can also be processed [35].

In this section the objectives of suggested scheme are narrowed to:

(a) Search for the next unlabeled pixel p.

(b) Flood-fill algorithm is used to label all the pixels in the connected component containing p.

(c) Repeat steps 1 and 2 until all the pixels are labeled.

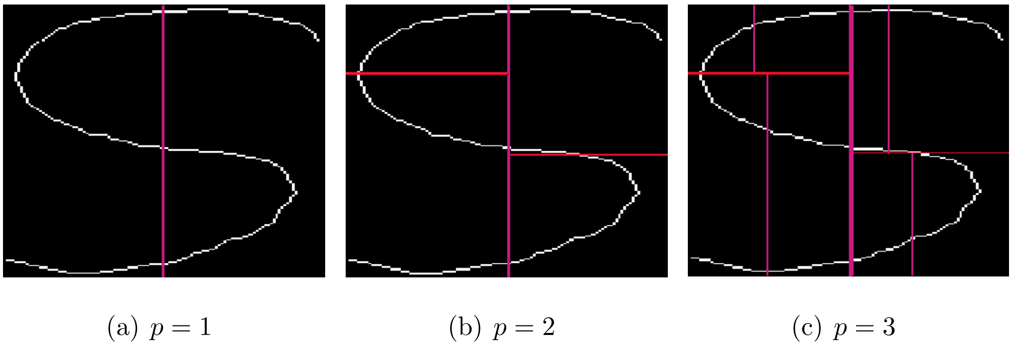
(d) Find the feature vector of labeled character.

4.1.1 Flood-Fill algorithm

The Flood Fill algorithm is to color an entire area of connected pixels with the same color [36]. It is an algorithm that determines the area connected to a given pixels in a multi-dimensional array. It is used in the "bucket" fill tool of paint programs to fill connected, similarly-colored areas with a different color. The flood fill algorithm takes three parameters: a start pixel, a target color, and a replacement color. The algorithm looks for all pixels in the array which are connected to the start node by a path of the target color, and changes them to the replacement color.

4.1.2 Feature Extraction

In this thesis, we have used geometrical property for feature vector representation of each numeral. These geometrical properties are extracted from binary image of the segmented numeral at different depth of spatial resolution through a hierarchical abstraction of the image data [37,38]. All images are scaled to a fixed size of $w \times w$ before feature extraction. For simulation purpose images are scaled to 128×128. The scaled image is then divided into sub-images, based on the k-d tree splitting strategy, which is a multi-dimensional binary search tree [37,39]. It is a recursive partitioning tree in which the partition is done along the x and y axis in alternative fashion. Each such partition of an image into two sub-images, define the depth of k-d tree decomposition. At each depth p total number of sub-image is 2^p as illustrated in Figure 4.1. The decomposition at $p = 1$ is done by dividing the image into two sub-images along the y-axis. Decomposition at subsequent depth are done by calling itself recursively on the transpose of each sub-images. Similarly splitting of image into sub-images at different depth is done along x-axis. Centroid of each sub-image is calculated relative to the centroid of the complete image and these values are normalized, dividing it by the number of rows or columns. These values are taken as a feature vector. At each depth p, the number of feature points is $2^{p+1} - 4$. The dimension of feature vector depends on the value of p.



(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

Figure 4.1: Image division based on K-d tree decomposition.

In the Figure 4.2, the numeral image is divided along y-axis at $p = 1$, gives two sub-images. At $p = 2$ the sub-images are divided along x-axis according to k-d tree decomposition. The result of such splitting gives the two feature point $y_0 - y_1$ and $y_0 - y_2$. Similarly, another two feature points $x_0 - x_1$ and $x_0 - x_2$ are obtained when the division of numeral image is done along x-axis at $p = 1$. Feature vector is normalized by dividing it by the number of rows or columns.

$$Feature\ vector = \left[\frac{y_0 - y_1}{w}, \frac{y_0 - y_2}{w}, \frac{x_0 - x_1}{w}, \frac{x_0 - x_2}{w}\right]$$

where w is the number of rows or columns.

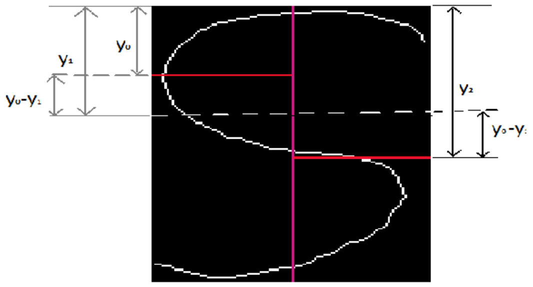


Figure 4.2: Illustration of feature vector for $p = 2$.

4.2 Recognition

Recognition is performed using minimum distance classifier. Any distance measure for the purpose of object matching should have the following properties: (a) It should have a large discriminatory power (b) its value should increase with the amount of difference between the two objects. One such distance measure is Modified Hausdorff Distance (MHD) [40] which is used for classification and recognition. For any given two set of points $A = (a_1, ..., a_{N_a})$ and $B = (b_1, ..., b_{N_b})$, MHD is given by,

$$\max(d(A, B), d(B, A))$$

where

$$d(A, B) = \frac{1}{N_a} \sum_{a \in A} d(a, B)$$

$$d(B, A) = \frac{1}{N_b} \sum_{b \in B} d(b, A)$$

$$d(a, B) = min_{b \in B} ||a - b||$$

$$d(b, A) = min_{a \in A} ||b - a||$$

N_a , N_b is number of element in A and B respectively. $\|a - b\|$ is the Euclidean distance between two points a and b.

4.2.1 Recognition Results

In this section the Receiver Operating Characteristic (ROC) has been drawn for English and Odia Numeral. To evaluate the performance of the proposed scheme, 10 samples of each English alphabet. Accuracy at different depth for English alphabet is shown in Figure 4.3. Maximum accuracy for English alphabet is 91% at depth 5.

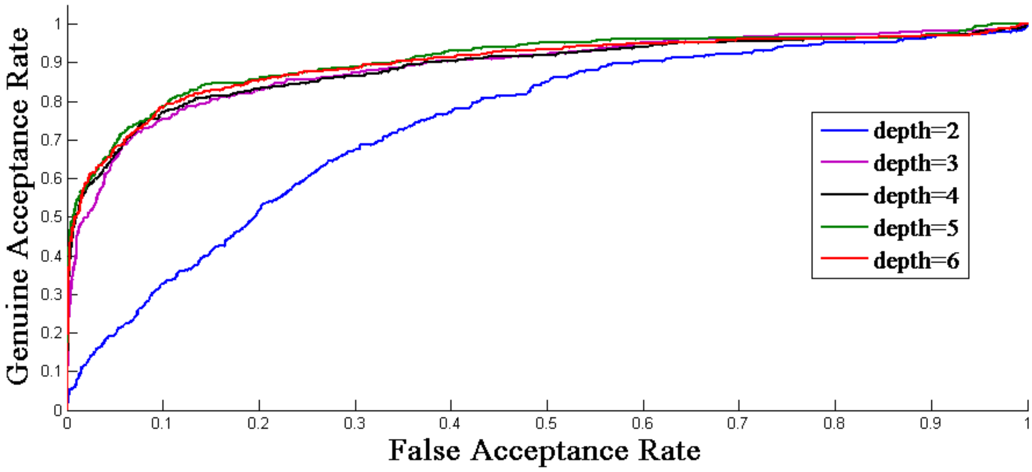


Figure 4.3: ROC at different depth for English numeral.

4.3 Summary

In this chapter each character image is extracted from hand written gesture and then centroid of sub-images are calculated relative to the centroid of complete image which is taken as a feature vector. Decomposition of image into sub-images are done using k-d tree splitting method. After feature vector computed, the train data set is classified by Modified Hausdorff Distance.

Chapter 5

Conclusion

In this thesis, we propose character extraction from hand gesture image, then feature extraction of that correspond character and recognition of alphabet specified through gesture. Motion of the color tip whose color i either red, green or blue is captured using a mobile camera having resolution 5M pixel. The video captured is in RGB color model which is converted into HSI color model and its motion is identified using optical flow method. Three different optical flow methods namely, Horn and Schunck, Lucas and Kanade, and Least Square Fit method are used, in which Horn and Schunck optical flow method has been shown to give the better results. Motion History Image (MHI) template are generated to get the character. Different gesture have different thickness, so to bring uniformity thinning is performed and the unwanted parasitic components are removed. Further motion of color tip is captured for multiple gesture and the unwanted lines are removed by proposed algorithm by opticalflow method. Both for single and multile character consider for removing the unwanted lines. There are some lines appear during formation of English alphabet which is removed by proposed algorithm in optical flow method. And also same technique is applied for mutiple characters formation. After getting the preprocessed alphabetical gesture image, each character is extracted from gesture image by flood fill algorithm and that correspond character is divided into sub-images using k-d tree decomposition method. Centroid of the sub-images is calculated relative to the centroid of complete image which is taken as feature vector and Modified Hausdorff Distance classifier is used for classification and recognition. An audio clip of each alphabet is stored and the correspond character is played during classification.

Scope for Further Research

The indoor environment under the uniform illumination can be extended to the outdoor environment with non-uniform illumination.

Bibliography

[1] J.K. Aggarwal and Michael S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[2] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257 – 267, 2001.

[3] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311 – 324, 2007.

[4] Caifeng Shan, Yucheng Wei, Xianchao Qiu, and Tieniu Tan. Gesture recognition using temporal template based trajectories. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR), 2004.*, volume 3, pages 954 – 957. IEEE, 2004.

[5] Masumi Ishikawa and Hiroko Matsumura. Recognition of a hand-gesture based on self-organization using a data glove. In *6th International Conference on Neural Information Processing, ICONIP'99.*, volume 2, pages 739 – 745. IEEE, 1999.

[6] M. Ali Qureshi, Abdul Aziz, Muhammad Ammar Saeed, Muhammad Hayat, and Jam Shahid Rasool. Implementation of an efficient algorithm for human hand gesture identification. In *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International*, pages 1 – 5. IEEE, 2011.

[7] Myoung Kyu Sohn, Sang Heon Lee, Dong Ju Kim, Byungmin Kim, and Hyunduk Kim. 3D hand gesture recognition from one example. In *IEEE International Conference on Consumer Electronics (ICCE) 2013*, pages 171 – 172. IEEE, 2013.

[8] Md. Ahad, Atiqur Rahman, J.K. Tan, H.S. Kim, and S. Ishikawa. Temporal motion recognition and segmentation approach. *International Journal of Imaging Systems and Technology*, 19(2):91 – 99, 2009.

[9] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185 – 203, 1981.

[10] Pengyu Hong, Matthew Turk, and Thomas S Huang. Gesture modeling and recognition using finite state machines. In *4th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 410 – 415. IEEE, 2000.

[11] Jing Lei, Zixue Cheng, and Wang Junbo. A recognition method for one-stroke finger gestures using a mems 3D accelerometer. *IEICE transactions on information and systems*, 94(5):1062 – 1072, 2011.

[12] Cemil Oz and Ming C Leu. American sign language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*, 24(7):1204 – 1213, 2011.

[13] Nikhil R Pal and Sankar K Pal. A review on image segmentation techniques. *Pattern recognition*, 26(9):1277 – 1294, 1993.

[14] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100 – 132, 1985.

[15] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888 – 905, 2000.

[16] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital image processing using MATLAB*, volume 2. Gatesmark Publishing Knoxville, 2009.

[17] Christoph Stiller, Janusz Konrad, and Robert Bosch. Estimating motion in image sequences - a tutorial on modeling and computation of 2D motion. *IEEE Signal Processing Magazine*, 16, 1999.

[18] Robert William Gainer Hunt, Michael R Pointer, and Michael Pointer. *Measuring colour.* John Wiley & Sons, 2011.

[19] Cecilia Di Ruberto. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*, 37(1):21 – 31, 2004.

[20] Pierre Soille. *Morphological image analysis: principles and applications.* Springer-Verlag New York, Inc., 2003.

[21] Eli Saber and A Murat Tekalp. Integration of color, edge, shape, and texture features for automatic region-based image annotation and retrieval. *Journal of Electronic Imaging*, 7(3):684 – 700, 1998.

[22] L. Luccheseyz and S.K. Mitray. Color image segmentation: A state-of-the-art survey. *Proceedings of the Indian National Science Academy (INSA-A). Delhi*, 67(2):207 – 221, 2001.

[23] Luca Zappella, Xavier Lladó, and Joaquim Salvi. Motion segmentation: A review. In *Proceedings of the 2008 conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, pages 398 – 407. IOS Press, 2008.

[24] Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433 – 466, 1995.

[25] Nils Papenberg, Andrés Bruhn, Thomas Brox, Stephan Didas, and Joachim Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141 – 158, 2006.

[26] L Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):774 – 780, 2000.

[27] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43 – 77, 1994.

[28] Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *9th IEEE International Conference on Computer Vision*, pages 726 – 733. IEEE, 2003.

[29] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674 – 679, 1981.

[30] Bruce David Lucas. *Generalized image matching by the method of differences*. PhD thesis, Carnegie Mellon University, 1985.

[31] Gary R Bradski and James W Davis. Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, 13(3):174 – 184, 2002.

[32] James W Davis. Sequential reliable-inference for rapid detection of human actions. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04).*, pages 111 – 111. IEEE, 2004.

[33] Osama Masoud and Nikos Papanikolopoulos. A method for human action recognition. *Image and Vision Computing*, 21(8):729 – 743, 2003.

[34] Kamaljeet Kaur and Mukesh Sharma. A method for binary image thinning using gradient and watershed algorithm. *International Journal*, 3(1), 2013.

[35] Michael B Dillencourt, Hanan Samet, and Markku Tamminen. A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM (JACM)*, 39(2):253–280, 1992.

[36] Jukka Arvo, Mika Hirvikorpi, and Joonas Tyystjärvi. Approximate soft shadows win an image-space flood-fill algorithm. In *Computer Graphics Forum*, volume 23, pages 271–279. Wiley Online Library, 2004.

[37] Alan Sexton, Alison Todman, and Kevin Woodward. Font recognition using shape-based quad-tree and kd-tree decomposition. In *3rd International Conference on Computer Vision, Pattern Recognition and Image Processing*, pages 212 – 215, 2000.

[38] Georgios Vamvakas, Basilis Gatos, and Stavros J Perantonis. Handwritten character recognition through two-stage foreground sub-sampling. *Pattern Recognition*, 43(8):2807 – 2816, 2010.

[39] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509 – 517, 1975.

[40] M P Dubuisson and Anil K. Jain. A modified hausdorff distance for object matching. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 1, pages 566 – 568. IEEE, 1994.

Dissemination

Conference

1. Lalit Mohan Pradhan, Shree Prakash and Banshidhar Majhi, Character Recognition Through Gesture, *The Springer Conference on Frontiers in Inteligent Computing Theory and Application (communicated)*