

A. EYE DETECTION USING VARIANTS OF HOUGH TRANSFORM

B. OFF-LINE SIGNATURE VERIFICATION

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

**Bachelor of Technology
In
Electronics & Instrumentation Engineering**

Submitted By: -

KAUSHAL KUMAR DHURUW
Roll No. – 10507007

ASWIN KUMAR TIGGA
Roll No. - 10507009

Under the Guidance of **Dr. S. Meher**



**Department of Electronics & Communication
Engineering**

**National Institute of Technology
Rourkela
2009**

A. EYE DETECTION USING VARIANTS OF HOUGH TRANSFORM

B. OFF-LINE SIGNATURE VERIFICATION

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

**Bachelor of Technology
In
Electronics & Instrumentation Engineering**

Submitted By: -

KAUSHAL KUMAR DHURUW
Roll No. – 10507007

ASWIN KUMAR TIGGA
Roll No. - 10507009

Under the Guidance of **Dr. S. Meher**



**Department of Electronics & Communication
Engineering**

**National Institute of Technology
Rourkela
2009**



National Institute of Technology Rourkela

CERTIFICATE

This is to certify that the thesis entitled “***1. EYE DETECTION, USING VARIANTS OF HOUGH TRANSFORM 2. OFFLINE SIGNATURE VERIFICATION***” submitted by Sri Kaushal Kumar Dhruw, Roll No. 10507007 in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Electronics & Instrumentation Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date:

(Dr. S.MEHER)



National Institute of Technology Rourkela

CERTIFICATE

This is to certify that the thesis entitled “***1. EYE DETECTION, USING VARIANTS OF HOUGH TRANSFORM 2. OFFLINE SIGNATURE VERIFICATION***” submitted by Sri Aswin Kumar Tigga, Roll No. 10507009 in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Electronics & Instrumentation Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date:

(Dr. S.MEHER)

ACKNOWLEDGEMENT

The most pleasant point of presenting a thesis is the opportunity to thank those who have contributed their guidance & help to it. I am grateful to Deptt. Of Electronics & Communication Engineering, N.I.T Rourkela, for giving me the opportunity to undertake this project, which is an integral part of the curriculum in B.Tech programme at the National Institute of Technology, Rourkela.

I would like to acknowledge the support of every individual who assisted me in making this project a success & I would like to thank & express heartfelt gratitude for my project guide **Dr. S. Meher**, who provided me with valuable inputs at the critical stages of this project execution along with guidance, support & direction without which this project would not have taken shape.

I am also thankful to the staff of Deptt. Of Electronics & Communication Engineering, N.I.T Rourkela, for co-operating with me & providing the necessary resources during the course of my project.

DATE:

PLACE: ROURKELA

KAUSHAL KUMAR DHRUW

Roll No. 10507007

Electronics & Instrumentation Engineering

National Institute of Technology,

Rourkela - 769008

ACKNOWLEDGEMENT

The most pleasant point of presenting a thesis is the opportunity to thank those who have contributed their guidance & help to it. I am grateful to Deptt. Of Electronics & Communication Engineering, N.I.T Rourkela, for giving me the opportunity to undertake this project, which is an integral part of the curriculum in B.Tech programme at the National Institute of Technology, Rourkela.

I would like to acknowledge the support of every individual who assisted me in making this project a success & I would like to thank & express heartfelt gratitude for my project guide **Dr. S. Meher**, who provided me with valuable inputs at the critical stages of this project execution along with guidance, support & direction without which this project would not have taken shape.

I am also thankful to the staff of Deptt. Of Electronics & Communication Engineering, N.I.T Rourkela, for co-operating with me & providing the necessary resources during the course of my project.

DATE:

PLACE: ROURKELA

ASWIN KUMAR TIGGA

Roll No. 10507009

Electronics & Instrumentation Engineering

National Institute of Technology,

Rourkela-769008

ABSTRACT

PART (A): EYE DETECTION USING VARIANTS OF HOUGH TRANSFORM

Broadly eye detection is the process of tracking the location of human eye in a face image. Previous approaches use complex techniques like neural network, Radial Basis Function networks, Multi-Layer Perceptrons etc. In the developed project human eye is modeled as a circle (iris; the black circular region of eye) enclosed inside an ellipse (eye-lashes). Due to the sudden intensity variations in the iris with respect the inner region of eye-lashes the probability of false acceptance is very less. Since the image taken is a face image the probability of false acceptance further reduces. Hough transform is used for circle (iris) and ellipse (eye-lash) detection. Hough transform was the obvious choice because of its resistance towards the holes in the boundary and noise present in the image. Image smoothing is done to reduce the presence of noise in the image further it makes the image better for further processing like edge detection (Prewitt method). Compared to the aforementioned models the proposed model is simple and efficient. The proposed model can further be improved by including various features like orientation angle of eye-lashes (which is assumed constant in the proposed model), and by making the parameters adaptive.

PART (B): OFF-LINE SIGNATURE VERIFICATION

Hand-written signature is widely used for authentication and identification of individual. It has been the target for fraudulence ever since. A novel off-line signature verification algorithm has been developed and tested successfully. Since the hand-written signature can be random, because of presence of various curves and features, techniques like character recognition cannot be applied for signature verification. The proposed algorithm incorporates a soft-computing technique “CLUSTERING” for extraction of feature points from the image of the signature. These feature points or centers are updated using the clustering update equations for required number of times, then these acts as extracted feature points of the signature image. To avoid interpersonal variation 6 to 8 signature images of the same person are taken and feature points are trained. These trained feature points are compared with the test signature images and based on a specific threshold, the signature is declared original or forgery. This approach works well if there is a high variation in the original signature, but for signatures with low variation, it produces incorrect results.

CONTENTS:

<i>PART (A): EYE DETECTION USING VARIANTS OF HOUGH TRANSFORM</i>	13
Chapter 1: Introduction	14
1.1: Overview	15
1.2: Biometric Technology	16
1.3: EYE: The Perfect ID	16
1.4: Assumptions	17
Chapter 2: Methods of eye detection	19
2.1: Introduction	20
2.2: Template Matching Method	20
2.3: Using Projection Function	21
2.4: IR Method	21
2.5: Support Vector Machines	22
2.6: Modelling	22
Chapter 3: Fundamentals and Theory	23
3.1: Hough Transform	24
3.2: Image Smoothing	26
3.3: Edge detection	27

Chapter 4: Overview of Algorithm	29
4.1: Specific Details	30
4.2: Algorithm	31
Chapter 5: Source Code	32
5.1: Image Smoothing and Edge Detection	33
5.2: Circle Detection	34
5.3: Ellipse Detection	36
Chapter 6: Simulation Results	38
Chapter 7: Conclusion	49
7.1: Scope of Improvement	50
References	51
 PART (B): OFF-LINE SIGNATURE VERIFICATION	 52
Chapter 1: Introduction	53
1.1: Introduction	54
1.2: How is it Different from Character Recognition?	54
1.3: Types of Forgeries	55
Chapter 2: Methods of Off-Line Signature Verification	56
2.1: Template Matching	57
2.2: Hidden Markov Models	58
2.3: Structural Techniques	58

2.4: Feature based technique based on global feature	59
Chapter 3: Fundamentals and Theory	60
3.1: Clustering	61
3.2: Geometric Feature Extraction	62
3.3: Image Binarization	62
Chapter 4: Applied Algorithm	63
4.1: Applied Algorithm	64
Chapter 5: Source Code	66
Chapter 6: Simulation Results	75
Chapter 7: Conclusion	96
7.1: Conclusion	97
7.2: Future Work	92
References	93

PART (A)

EYE DETECTION, USING VARIANTS OF HOUGH TRANSFORM

CHAPTER 01:

INTRODUCTION

1.1. Overview:

One of the basic problems in the design of an object recognition system is the selection of a set of statistical features to be extracted from the original image for the purpose of classification. Without classification it is very difficult to recognize any object. Objects are recognized from many different vantage points (from the front, side, or back), in many different places, and in different sizes. Objects can even be recognized when they are partially obstructed from view.

For any object in an image, there are many 'features' which are interesting points on the object that can be extracted to provide a "feature" description of the object.

The goal of computer-based object recognition is to identify objects in images, and if necessary to determine their three-dimensional location and orientation. Objects are identified by comparing data extracted from images to a priori models of objects or object classes in memory. When a model matches the image data, an object is recognized, and features of the object, including its location and orientation in the world (i.e. its pose), can be recovered from the data-to-model correspondence.

We humans can easily characterize object by identifying object boundaries or edges. Edges characterize objects, so edge is an important term in relation to object recognition. Our project consists of detection of human eye (i.e. extraction of its location and orientation information) in a grayscale image. There has been a lot of active research in the area, with algorithms based on Kalman filtering, eigenface decomposition, but also more complex techniques, employing multiple stages of Bayesian classification, clustering and post processing or updated template matching. Many of these approaches are very computationally intensive (requiring neural network trainings or large amounts of parallel processing), and many need color information. Various techniques have been proposed using texture, depth, shape and color information or combinations of these for eye detection. Robust, accurate and non-intrusive eye detection and tracking mechanism remains a largely unresolved problem.

1.2. Biometric Technology:

A biometric system provides automatic recognition of an individual based on some sort of unique feature or characteristic possessed by the individual. Biometric systems have been developed based on fingerprints, facial features, voice, hand geometry, handwriting, the retina, and the one presented in this project, the eye.

Biometric systems work by first capturing a sample of the feature, such as recording a digital sound signal for voice recognition, or taking a digital colour image for eye detection. The sample is then transformed using some sort of mathematical function into a biometric template. The biometric template will provide a normalized, efficient and highly discriminating representation of the feature, which can then be objectively compared with other templates in order to determine identity.

Most biometric systems allow two modes of operation. A training mode or enrolment mode for adding templates to a database, and an identification mode, where a template is created for an individual and then a match is searched for in the database of pre-enrolled templates. A good biometric is characterized by use of a feature that is; highly unique so that the chance of any two people having the same characteristic will be minimal, stable so that the feature does not change over time, and be easily captured in order to provide convenience to the user, and prevent misrepresentation of the feature.

1.3. EYE: The Perfect ID

The randomness and uniqueness of human eye patterns is a major breakthrough in the search for quicker, easier and highly reliable forms of automatic human identification, where the human eye serves as a type of 'biological passport, PIN or password'.

Results of a study by John Daugman and Cathryn, of over two million different pairs of human eyes in images taken from volunteers in Britain, USA and Japan show that no two eye patterns were the same in

even as much as one-third of their form. Even genetically identical faces - for example from twins or in the probable future, from human clones - have different eye patterns.

The implications of eye detection are highly significant at a time when organizations such as banks and airlines are looking for more effective security measures. The possible applications of eye detection span all aspects of daily life, from computer login, national border controls and secure access to bank cash machine accounts, to ticket-less air travel, access to premises such as the home and office, benefits entitlement and credit card authentication.

Compared with other biometric technologies, such as face, speech and finger recognition, eye recognition can easily be considered as the most reliable form of biometric. However, there have been no independent trials of the technology, and source code for systems is not available in working condition.

1.4. Assumptions:

For human eye detection we took the following assumptions: -

- Image should not be too noisy.
- Eyes should be in normal horizontal position (i.e. the head should not be tilted).
- Iris diameter shouldn't be very small in respect to the size of image.
- The two candidate circles for irises must have similar radiuses. For a normal human subject, the irises are not that different, and have a diameter of around 12 mm (the normal human pupil is around 2-3 mm in daylight and can go to 7mm during nighttime). Therefore, if the difference between the radiuses exceeds a certain threshold, we can rule out the pair of candidates.
- The distance between the two centers of the circles divided by the average radius of one circle had to be bounded between some values. In a normal human, the inter eye distance is of about 63 mm;

with an iris diameter of about 12 mm, this means that the value of the fraction has to be around 5.

We could also use the distance between the centers of the ellipses for similar considerations.

Since the ellipse tracking mechanism only searched in an area surrounding the irises, the two obtained ellipses (“eyelashes”) had to have similar orientation. When the differences in orientation were too large, the candidates will be ruled out.

CHAPTER 02:

METHODS OF EYE DETECTION

2.1. INTRODUCTION:

A lot of research work has been published in the field of eye detection in the last decade. Various techniques have been proposed using texture, depth, shape and colour information or combinations of these for eye detection. Vezhnevets focus on several landmark points (eye corners, iris border points), from which the approximate eyelid contours are estimated. The upper eyelid points are found using on the observation that eye border pixels are significantly darker than surrounding skin and sclera. The detected eye boundary points are filtered to remove outliers and a polynomial curve is fitted to the remaining boundary points. The lower lid is estimated from the known iris and eye. Some of the famous eye detection techniques are discussed below.

2.2. TEMPLATE MATCHING METHOD:

Reinders present a method where based on the technique of template matching the positions of the eyes on the face image can be followed throughout a sequence of video images. Template matching is one of the most typical techniques for feature extraction. Correlation is commonly exploited to measure the similarity between a stored template and the window image under consideration. Templates should be deliberately designed to cover variety of possible image variations. During the search in the whole image, scale and rotation should also be considered carefully to speed up the process. To increase the robustness of the tracking scheme the method automatically generates a codebook of images representing the encountered different appearances of the eyes. Yuille first proposed using deformable templates in locating human eye. The weaknesses of the deformable templates are that the processing time is lengthy and success relies on the initial position of the template. Lam introduced the concept of eye corners to improve the deformable template approach.

2.3. USING PROJECTION FUNCTION:

Saber and Jeng proposed to use facial features geometrical structure to estimate the location of eyes. Takacs developed iconic filter banks for detecting facial landmarks. projection functions have also been employed to locate eye windows. Feng and Yeun developed a variance projection function for locating the corner points of the eye. Zhou and Geng propose a hybrid projection function to locate the eyes.

By combining an integral projection function, which considers mean of intensity, and a variance projection function, which considers the variance of intensity, the hybrid function better captures the vertical variation in intensity of the eyes. Kumar suggest a technique in which possible eye areas are localized using a simple thresholding in colour space followed by a connected component analysis to quantify spatially connected regions and further reduce the search space to determine the contending eye pair windows. Finally the mean and variance projection functions are utilized in each eye pair window to validate the presence of the eye. Feng and Yeun employ multi cues for eye detection on gray images using variance projection function.

2.4. IR METHOD:

The most common approach employed to achieve eye detection in real-time is by using infrared lighting to capture the physiological properties of eyes and an appearance-based model to represent the eye patterns. The appearance-based approach detects eyes based on the intensity distribution of the eyes by exploiting the differences in appearance of eyes from the rest of the face. This method requires a significant number of training data to enumerate all possible appearances of eyes i.e. representing the eyes of different subjects, under different face orientations, and different illumination conditions.

The collected data is used to train a classifier such as a neural net or support vector machine to achieve detection.

2.5. SUPPORT VECTOR MACHINES (SVMs).

Support Vector Machines (SVMs) have been recently proposed by Vapnik and his co-workers as a very effective method for general-purpose pattern recognition.

Intuitively, given a set of points belonging to two classes, a SVM finds the hyper-plane that separates the largest possible fraction of points of the same class to the same side while maximizing the distances from either class to the hyper-plane. This hyper-plane is called Optimal Separating Hyper-plane (OSH). It minimizes the risk of misclassifying not only the samples in the training set but also the unseen samples in the test set. The application of SVMs to computer vision area has emerged recently. Osuna train a SVM for face detection, where the discrimination is between two classes: face and non-face, each with thousands of samples. Guo and Stan show that the SVMs can be effectively trained for face recognition and is a better learning algorithm than the nearest centre approach.

(etc...)

2.6. MODELLING:

This is the simplest yet efficient method for eye detection. We modelled the human eye as a circle circumscribed in an ellipse, where circle represents the iris of human eye and the ellipse represents the eye lashes. This is not the only model for an eye; there are infinite models possible and they can be applied as per requirement. Hough Transform can be used for the detection of aforesaid circle and ellipse then final eye is detected by neglecting the wrong detections and ruling out a pair of eyes based on geometrical considerations. This method is applied for offline eye detection purpose.

CHAPTER 03:

FUNDAMENTALS AND THEORY

3.1. HOUGH TRANSFORM:

The Hough Transform is an algorithm presented by Paul Hough in 1962 for the detection of features of a particular shape like lines or circles in digitalized images. In its classical form it was restricted to features that can be described in a parametric form. However, a generalized version of the algorithm exists which can also be applied to features with no simple analytic form, but it is very complex in terms of computation time.

Hough transform can be applied to many computer vision problems as most images contain feature boundaries which can be described by regular curves. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise, unlike edge detectors.

One important difference between the Hough transform and other approaches is resistance of the former to noise in the image and its tolerance towards holes in the boundary line.

3.1.1. LINE DETECTION:

The simplest case of Hough transform is the linear transform for detecting straight lines. In the image space, the straight line can be described as

$$y = mx + b$$

And can be graphically plotted for each pair of image points (x, y). If the line is horizontal, then a is 0, and if the line is vertical, then a is infinite. So, a more general representation of a line will be

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta$$

3.1.2. CIRCLE DETECTION:

Hough transform can also be used for detection of circles and other parametrizable geometric figures. In theory any kind of curve can be detected if you can express it as a function of the form

$$f(a_1, a_2, \dots, a_n, x, y) = 0.$$

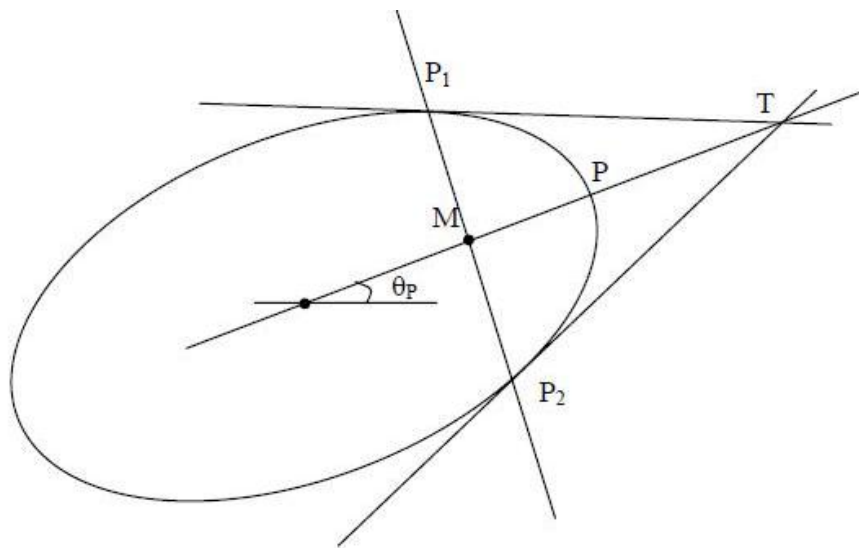
For example a circle can be represented as

$$(x - a)^2 + (y - b)^2 - r^2 = 0$$

Then we have a n dimensional parameter space (three dimensional space for a circle). This model has three parameters: two parameters for the centre co-ordinates of the circle and one parameter for the radius of the circle.

3.1.3. ELLIPSE DETECTION:

The strategy is to first construct the ellipse center from the tangent directions of a number of edge points, making use of the geometry of an ellipse.



An ellipse can be represented in parametric form as:

$$x(\theta) = a_0 + a_x \sin(\theta) + b_x \cos(\theta)$$

$$y(\theta) = b_0 + a_y \sin(\theta) + b_y \cos(\theta)$$

Where (a_0, b_0) is the center of the ellipse and (a_x, a_y) and (b_x, b_y) are vectors representing the major and minor axes of the ellipse.

3.2 IMAGE SMOOTHING:

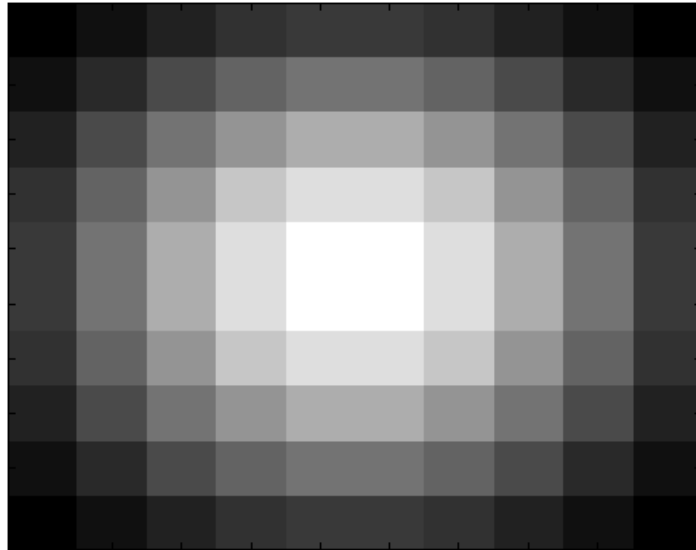
Gaussian low pass filters are used for smoothing the image. The form of these filters in 2-dimensions is given by

$$H(u, v) = \exp(-(D(u, v)^2)/(2\sigma^2))$$

Where $D(u, v)$ is the distance from the origin of the Fourier transform. ' σ ' is a measure of the spread of the Gaussian curve. The following matrix is used for smoothing:

$$\frac{1}{64} \begin{vmatrix} 0 & 2 & 4 & 2 & 0 \\ 2 & 2 & 4 & 2 & 2 \\ 4 & 4 & 8 & 4 & 4 \\ 2 & 2 & 4 & 2 & 2 \\ 0 & 2 & 4 & 2 & 0 \end{vmatrix} .$$

Approximate image form of the matrix used for smoothing:



3.3 EDGE DETECTION:

Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames as a precursor step to feature extraction and object segmentation. This process detects outlines of an object and boundaries between objects and the background in the image. An edge-detection filter can also be used to improve the appearance of blurred or anti-aliased video streams.

The basic edge-detection operator is a matrix-area gradient operator that determines the level of variance between different pixels. The edge-detection operator is calculated by forming a matrix centered on a pixel chosen as the center of the matrix area. If the value of this matrix area is above a given threshold, then the middle pixel is classified as an edge. Examples of gradient-based edge detectors are Roberts, Prewitt, and Sobel operators. All the gradient-based algorithms have kernel operators that

calculate the strength of the slope in directions which are orthogonal to each other, commonly vertical and horizontal. Later, the contributions of the different components of the slopes are combined to give the total value of the edge strength.

We used Prewitt operator to detect the edges. Prewitt operator is implemented using the following two functions:

$$S_x = (a_2 + c \cdot a_3 + a_4) - (a_0 + c \cdot a_7 + a_6)$$

$$S_y = (a_0 + c \cdot a_1 + a_2) - (a_6 + c \cdot a_5 + a_4)$$

Or in matrix form S_x and S_y can be represented as:

$$S_y = \begin{vmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{vmatrix} \quad S_x = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{vmatrix}$$

$$M(i, j) = [S_x^2 + S_y^2]^{(1/2)}$$

CHAPTER 04:

OVERVIEW OF ALGORITHM

4.1. SPECIFIC DETAILS:

In our Project we have modeled the human as a circle circumscribed in an ellipse as discussed earlier, where the circle represents the iris of the human eye and the elliptical part represents the eye lashes, and then the Hough transform is used to detect this modeled eye from a given image and then final pair of eye is given by considering the geometry of the human eye.

The Image of interest contains many features and noise that are completely not of interest, so to remove or avoid them smoothing of the image is done. As we know that the color of the eye inside eye-lash is white, which makes the eyes significant and prominent as the face color is quite dull in comparison to the pure white color.

Due to the prominent color of eye the image smoothing using Gaussian operator has no significant effect on the edges and on the iris of the human eye.

Edge detection is performed to further avoid the features and details that are not of interest. A log-linear transform can also be performed for the aforementioned operation. Edge detected where there is a sudden intensity change based on a pre-specified threshold used in the edge-detection by Prewitt operator. The area of almost constant intensity is left white, while the edges are marked black. This processed image then undergoes Hough transform where circles and ellipses are detected.

It is possible that this algorithm may detect circles other than the iris of the human eye and ellipses very different from the eye lashes. So, final detection is done based on geometrical considerations of human eye. Like ratio of iris diameter to the eye-lash major axis is constant and ratio of iris diameter to the distance between the centers of the two iris is constant etc.

The stepwise algorithm is summarized below: -

4.2. ALGORITHM:

1. Take input the image of which eye detection has to be performed.
2. Convert color image into grayscale image.
3. Smoothen the image to remove noise present in the image and to avoid any feature that is not of interest using Gaussian mask as described in Chapter 03
4. Detect the edges of the image using the Prewitt operator as discussed in chapter 03.
5. Detect the iris (black circular region inside eye) using Hough transform for circle detection.
6. Store the center and radii of the circles thus detected from the previous step.
7. Detect the presence of the ellipses surrounding the circle again using the Hough transform.
8. Geometrically match each of the detected circle enclosed in an ellipse
9. Rule out the pair of eyes based on geometrical considerations from the above step.

CHAPTER 05:

SOURCE CODE

5.1. IMAGE SMOOTHING AND EDGE DETECTION

```
clear all;
close all;
clc;

msk = [0      1/32      1/16      1/32      0;
       1/32      1/32      1/16      1/32      1/32;
       1/16      1/16      1/8      1/16      1/16;
       1/32      1/32      1/16      1/32      1/32;
       0      1/32      1/16      1/32      0];

%msk = ones(5,5);
%msk = msk.*(1/25);

img_prt = double(zeros(5,5));
pic_tot = 0;
%x = imread('D:\Study\Project 2008-
09\downloads\Eye_tracking\Eye_tracking\jpg\k091.jpg');
x = rgb2gray(imread('C:\Users\Kaushal\Pictures\t3.jpg'));
x2 = x;
%x = rgb2gray(imread('D:\wallpapers\ELISHA CUTHBERT\34.jpg'));
%x = rgb2gray(x);

[m,n] = size(x);
I1 = edge(x, 'prewitt');
imshow(I1);
y2 = x;

for k = 1:1:5
    for i=1:1:(m-4)
        for j = 1:1:(n-4)

            img_prt = double(y2(i:i+4,j:j+4));
            img_prt = img_prt.*msk;
            pic_tot = uint8(round(sum(img_prt(:)))));
            y(i,j) = pic_tot;
        end
    end
    [m n] = size(y);
    y2 = y;
end
```

```
figure;
imshow(y);
y1 = edge(y, 'prewitt');
figure;
imshow(y1);
```

5.2. CIRCLE DETECTION

```
clear all;
close all;
clc;

x = rgb2gray(imread('C:\Users\Kaushal\Pictures\t3.jpg'));

x2 = x;
%x = rgb2gray(imread('D:\wallpapers\ELISHA CUTHBERT\34.jpg'));
%x = rgb2gray(x);

figure;
imshow(y);
y1 = edge(x, 'prewitt');
figure;
imshow(y1);

% imshow(x);
% figure; imshow(y);
% for i=1:1:(m-4)
%     for j = 1:1:(n-4)
%
%         img_prt = double(y(i:i+4,j:j+4));
%         img_prt = img_prt.*msk;
%         pic_tot = uint8(round(sum(img_prt(:))));
%         zc(i,j) = pic_tot;
%
%     end
% end
%I = rgb2gray(imread('D:\wallpapers\ELISHA CUTHBERT\34.jpg'));
%I =
rgb2gray(imread('C:\Users\Kaushal\Pictures\test_sub002.jpg'));
%I = x;
```

```

%I = edge(I,'prewitt');
%I =im2bw(I);
%[y,x]=find(I);
%[sy,sx]=size(I);
%imshow(I);

I = y1;

[y,x]=find(I);
[sy,sx]=size(I);
figure;imshow(I);

totalpix = length(x);

cntdn = 0;

HM = zeros(sy,sx,50);
R = 1:50;
R2 = R.^2;
sz = sy*sx;

for cnt = 1:totalpix
for cntR = 1:50
b = 1:sy;
a = (round(x(cnt) - sqrt(R2(cntR) - (y(cnt) - [1:sy]).^2)));
b = b(imag(a)==0 & a>0);
a = a(imag(a)==0 & a>0);
ind = sub2ind([sy,sx],b,a);
HM(sz*(cntR-1)+ind) = HM(sz*(cntR-1)+ind) + 1;
%disp(ind);
end
%cntdn = cntdn + 1
end

for cnt = 1:50
H(cnt) = max(max(HM(:, :, cnt)));
end
figure;
plot(H, '*-');
[maxval, maxind] = max(H);
[B,A] = find(HM(:, :, maxind)==maxval);
figure;
imshow(I); hold on;
plot(mean(A),mean(B), 'xr')
text(mean(A),mean(B),num2str(maxind), 'color', 'green')

```

5.3. ELLIPSE DETECTION

```
clear all;
close all;
clc;

%I = rgb2gray(imread('D:\wallpapers\ELISHA CUTHBERT\34.jpg'));
I =
rgb2gray(imread('C:\Users\Kaushal\Pictures\test_img010.jpg'));
%I = edge(I, 'roberts');
I =im2bw(I);
[y,x]=find(I);
[sy,sx]=size(I);
imshow(I);

totalpix = length(x);

cntdn = 0;

HM = zeros(sy,sx,50);
%R = 1:50;
%R2 = R.^2;
a1 = 101:150;
a2 = a1.^2;
b1 = 51:100;
b2 = b1.^2;
sz = sy*sx;

for cnt = 1:totalpix
for cntR = 1:50
k = 1:sy;
%a = (round(x(cnt) - sqrt(R2(cntR) - (y(cnt) - [1:sy]).^2)));
h = (round(x(cnt) - a2(cntR).*sqrt(1 - ((y(cnt)-
[1:sy])./b2(cntR)).^2)));

k = k(imag(h)==0 & h>0);
h = h(imag(h)==0 & h>0);
ind = sub2ind([sy,sx],k,h);
HM(sz*(cntR-1)+ind) = HM(sz*(cntR-1)+ind) + 1;
end
cntdn = cntdn + 1
end

for cnt = 1:50
H(cnt) = max(max(HM(:, :, cnt)));
end
```

```

figure;
plot(H, '*-');
[maxval, maxind] = max(H);
[B,A] = find(HM(:, :, maxind) == maxval);
figure;
imshow(I); hold on;
plot(mean(A), mean(B), 'xr')
%text(mean(A), mean(B), num2str(maxind), 'color', 'green')
text(mean(A), mean(B), num2str(maxind), 'color', 'green')

```

CHAPTER 06:

SIMULATION RESULTS

Test subject # 01

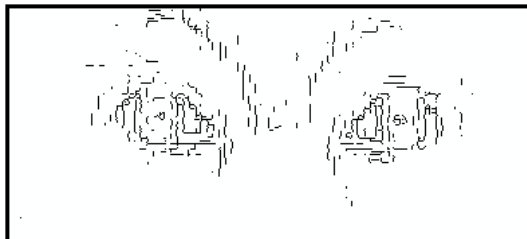
ORIGINAL IMAGE



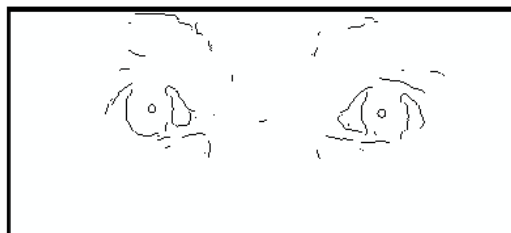
SMOOTHED IMAGE



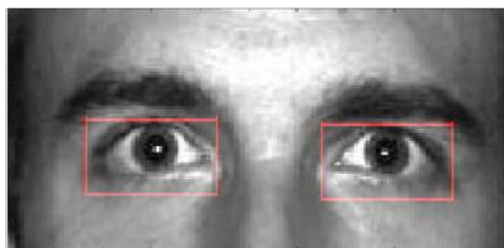
EDGES OF ORIGINAL IMAGE



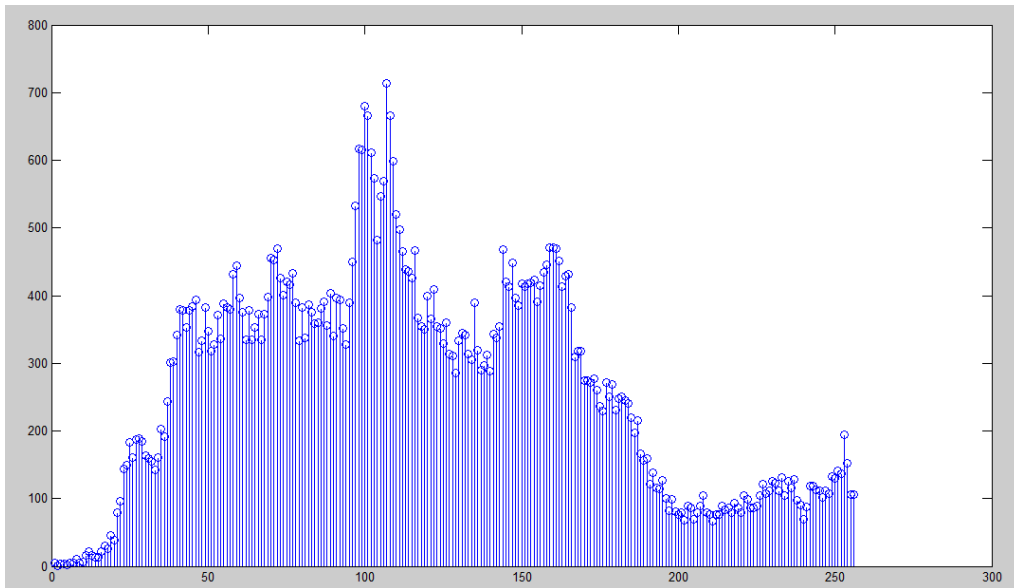
EDGES OF SMOOTHED IMAGE



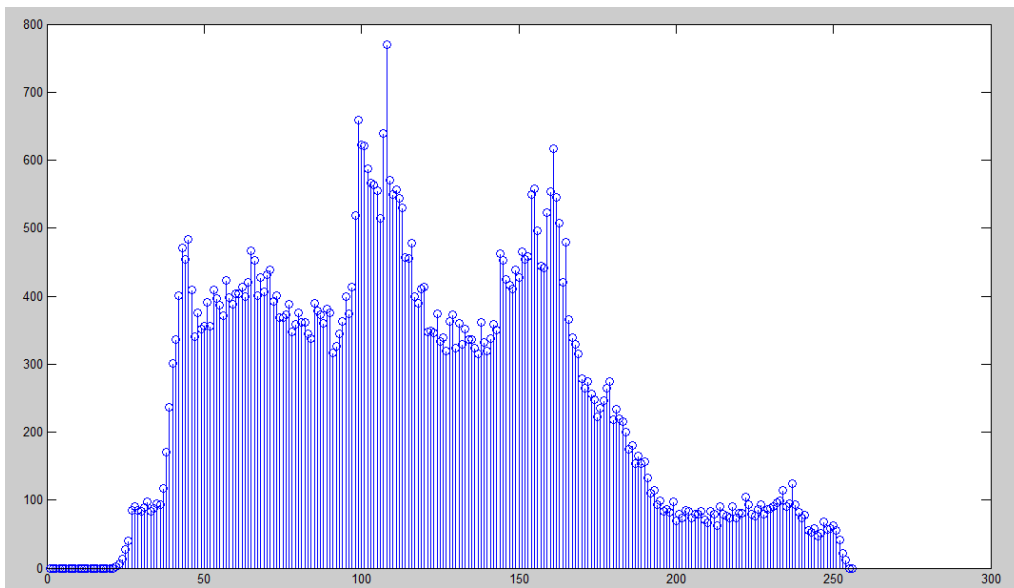
DETECTED EYES



HISTOGRAM OF ORIGINAL IMAGE



HISTOGRAM OF SMOOTHED IMAGE



Co-ordinates of centre of left iris = (70,103). Radius of left iris = 31

Co-ordinates of centre of right iris = (71,293). Radius of right iris = 31

Test subject # 02

ORIGINAL IMAGE



SMOOTHED IMAGE



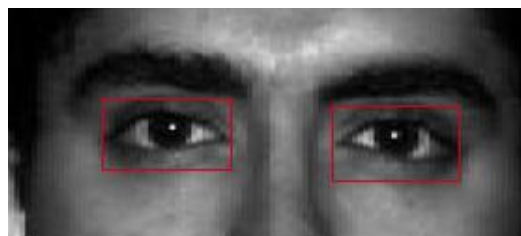
EDGES OF ORIGINAL IMAGE



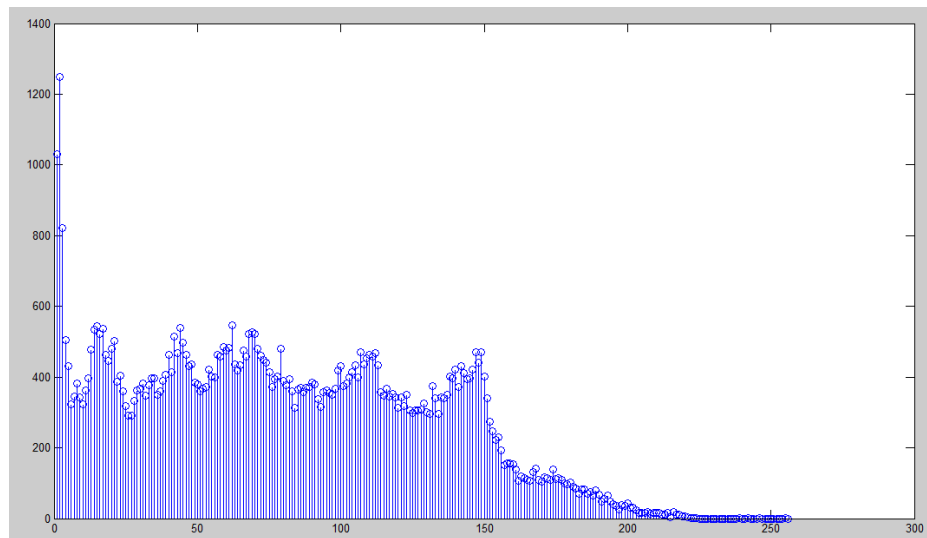
EDGES OF SMOOTHED IMAGE



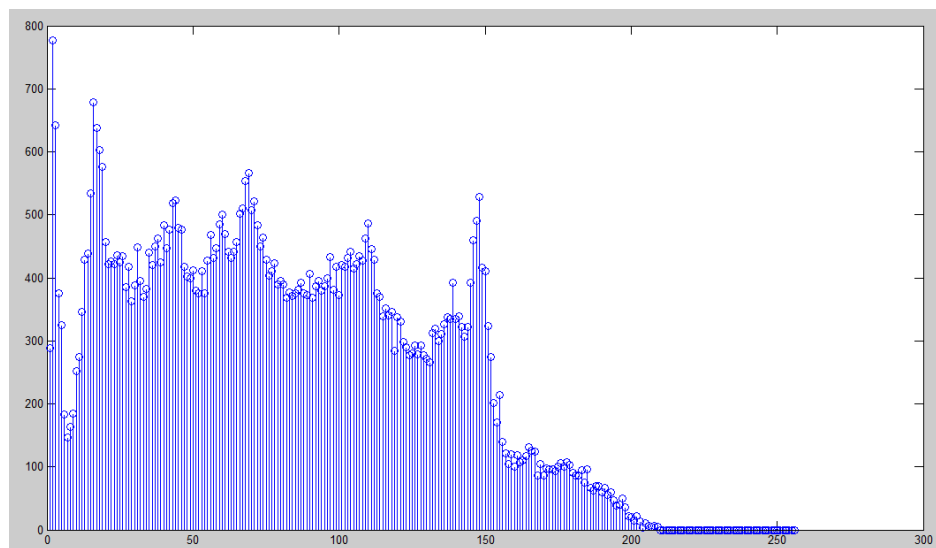
DETETED EYES



HISTOGRAM OF ORIGINAL IMAGE



HISTOGRAM OF SMOOTHED IMAGE



Co-ordinates of centre of left iris = (84, 97). Radius of left iris = 34

Co-ordinates of centre of right iris = (86,291). Radius of right iris = 36

Test subject # 03

ORIGINAL IMAGE



SMOOTHED IMAGE



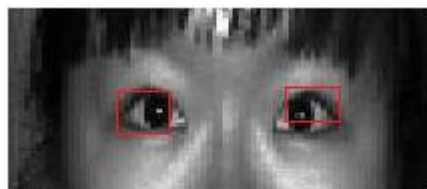
EDGES OF ORIGINAL IMAGE



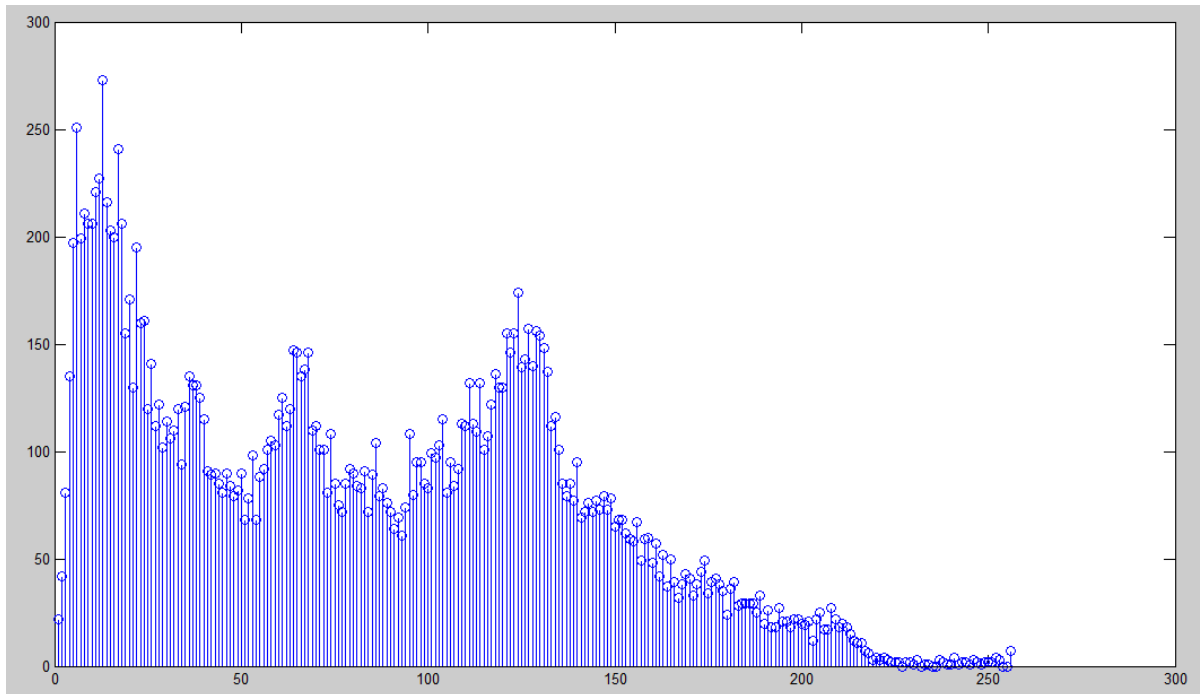
EDGES OF SMOOTHED IMAGE



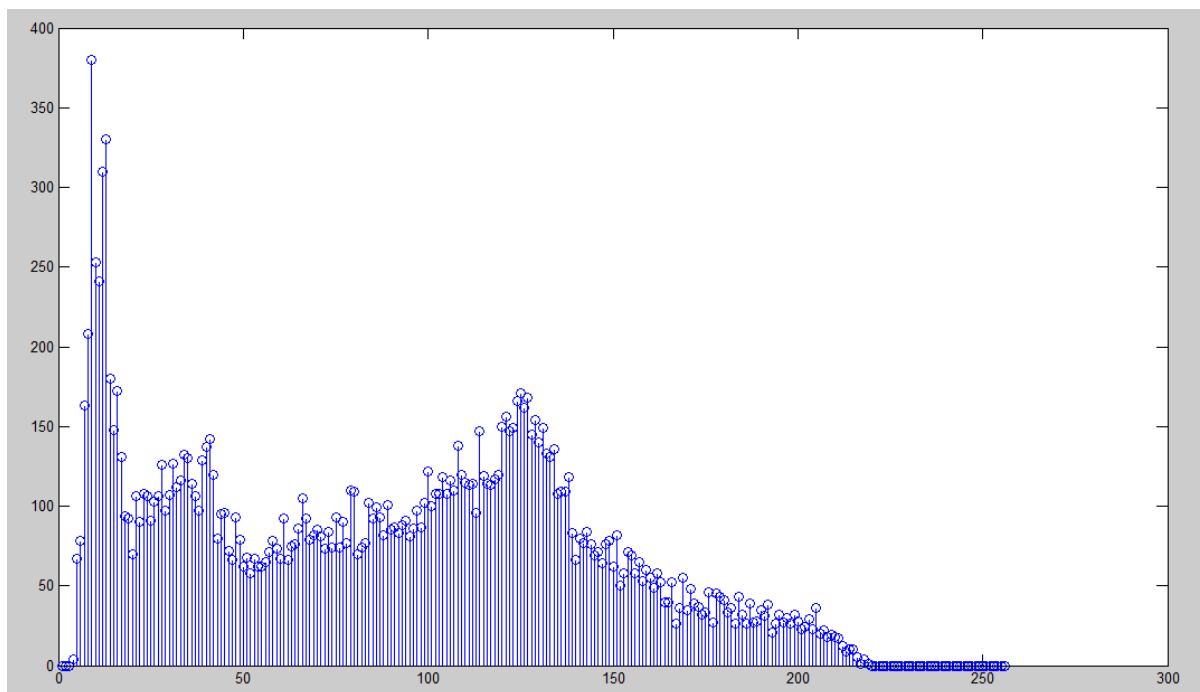
DETECTED EYES



HISTOGRAM OF ORIGINAL IMAGE



HISTOGRAM OF SMOOTHED IMAGE



Co-ordinates of centre of left iris = (40, 56). Radius of left iris = 16

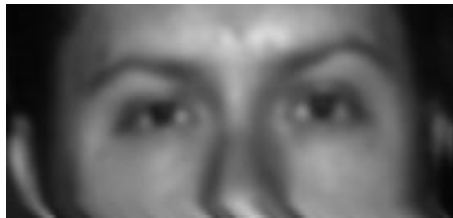
Co-ordinates of centre of right iris = (42,133). Radius of right iris = 17

Test subject # 04

ORIGINAL IMAGE



SMOOTHED IMAGE



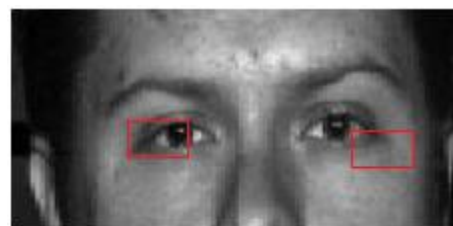
EDGES OF ORIGINAL IMAGE



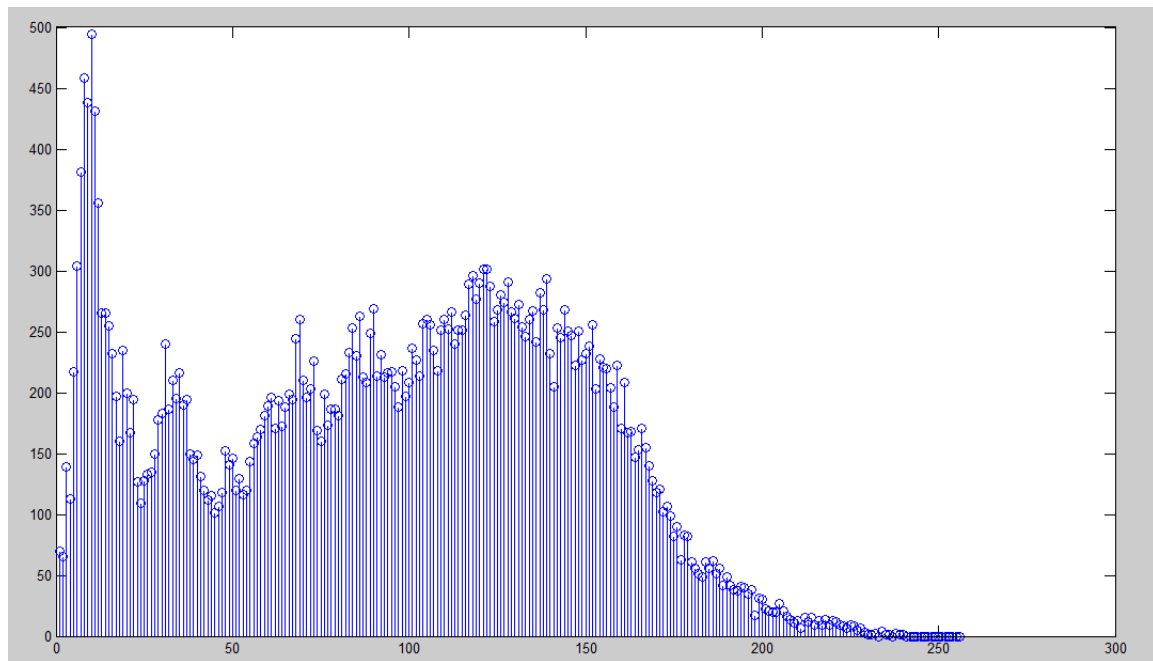
EDGES OF SMOOTHED IMAGE



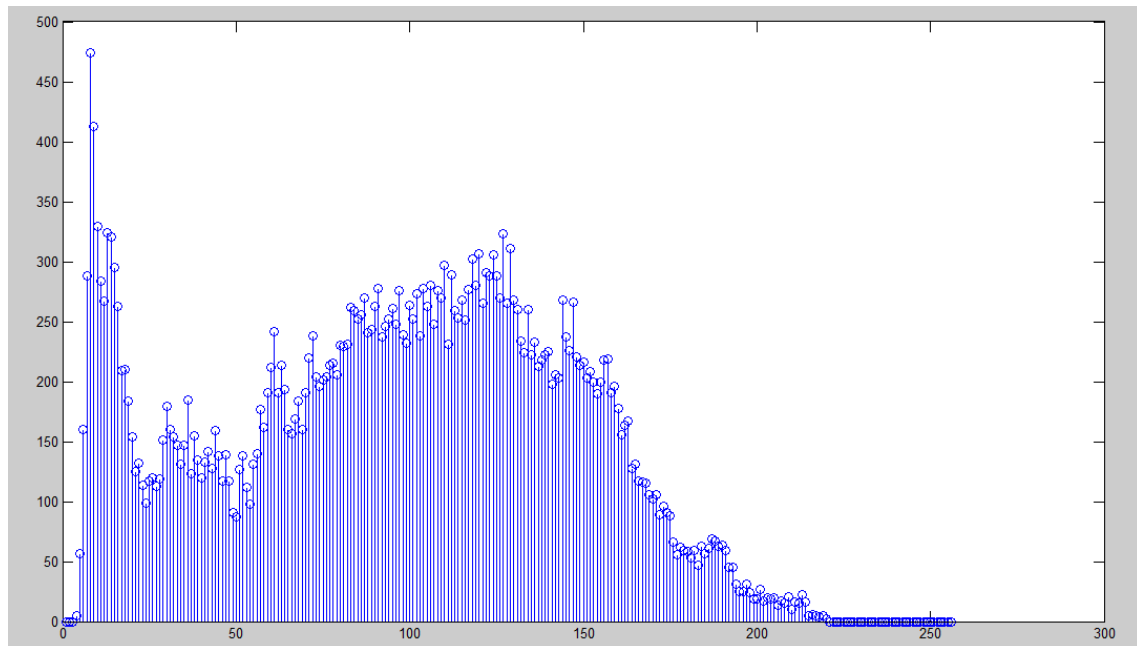
DETECTED EYE



HISTOGRAM OF ORIGINAL IMAGE



HISTOGRAM OF SMOOTHED IMAGE



Co-ordinates of centre of left iris = (73, 98). Radius of left iris = 27

Co-ordinates of centre of right iris = (79, 196). Radius of right iris = 27

Test subject # 05

ORIGINAL IMAGE



SMOOTHED IMAGE



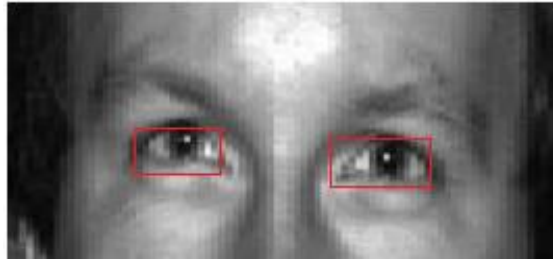
EDGES OF ORIGINAL IMAGE



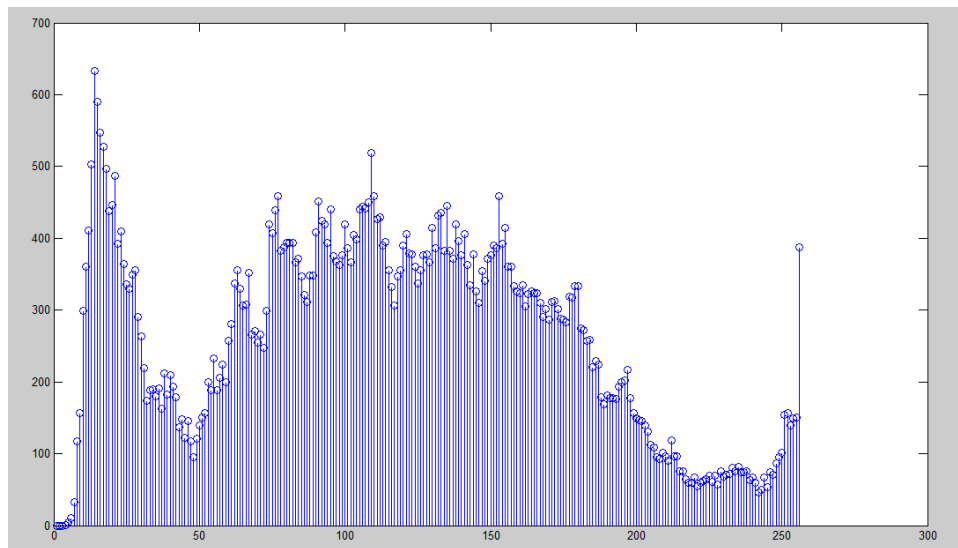
EDGES OF SMOOTHED IMAGE



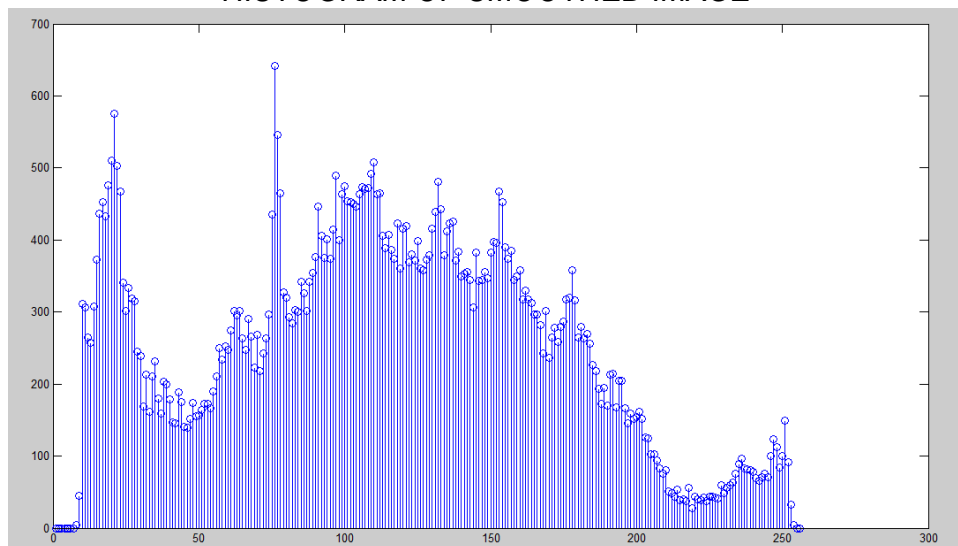
DETECTED EYES



HISTOGRAM OF ORIGINAL IMAGE



HISTOGRAM OF SMOOTHED IMAGE



Co-ordinates of centre of left iris = (93, 88). Radius of left iris = 32

Co-ordinates of centre of right iris = (96,247). Radius of right iris = 31

CHAPTER 07:

CONCLUSION

In this thesis we discussed the detection of human eye in still images. We discussed about the preprocessing required, which includes RGB to gray level conversion, image smoothing and Prewitt edge detection. Finally the Hough transform is applied for detection of circles and ellipses and pair of detected eyes are ruled out by geometrical considerations. The type of approach discussed here cannot be applied for real-time eye detection scheme; however it has been found that this technique efficiently detects eyes in still images and can be used for off-line eye detection system. The results of the discussed algorithm have been found satisfactory. Hough transform has been used because of its robustness for noise and resistance towards discontinuity in standard geometrical structures.

7.1. SCOPE OF IMPROVEMENT:

- A better human eye model can be used for making the algorithm more robust and for reducing the false acceptance ratio.
- A better low pass filter or an adaptive low pass filter can be used for exact removal of noise in the image as well as for making the image better for further processing. (e. g. edge detection etc.)
- Prewitt edge detection sometimes detects edges that are not of interest because of its pre-specified threshold. This threshold can be made adaptive that will change according to the image for a better edge detection.

REFERENCES:

1. Digital Image Processing By Gonzalez and Woods (Prentice Hall).
2. DIP and Analysis By B.Chandra and D.Dutta Majumdar.
3. K. M. Lam, H. Yan, " Locating and extracting the eye in human face images", Pattern Recognition, Vol. 29, No. 5 pp.771-779.(1996).
4. Kumar, Thilak R and Raja, Kumar S and Ramakrishnan, "Eye detection using colour cues and projection functions", Proceedings 2002 International Conference on Image Processing ICIP, pages Vol.3 337-340, Rochester, New York, USA.
5. L. Ma, Y. Wang, T. Tan. Iris recognition using circular symmetric filters. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 2002.

PART (B)

OFF-LINE SIGNATURE VERIFICATION

CHAPTER 01:

INTRODUCTION

1.1. INTRODUCTION: -

Handwritten signature is one of the most widely accepted personal attributes for identity verification. As a symbol of consent and authorization, especially in the prevalence of credit cards and bank cheques, handwritten signature has long been the target of fraudulence. Therefore, with the growing demand for processing of individual identification faster and more accurately, the design of an automatic signature system faces a real challenge.

Handwritten signature verification can be divided into on-line (or dynamic) and off-line (or static) verification. On-line verification refers to a process that the signer uses a special pen called a stylus to create his or her signature, producing the pen location, speeds and pressures, while off-line verification just deals with signature images acquired by a scanner or a digital camera.

In an *off-line* signature verification system, a signature is acquired as an image. This image represents a personal style of human handwriting, extensively described by the graphometry. In such a system the objective is to detect different types of forgeries, which are related to intra and inter-personal variability. The system applied should be able to overlook inter-personal variability and mark these as original and should be able to detect intra-personal variability and mark them as forgeries.

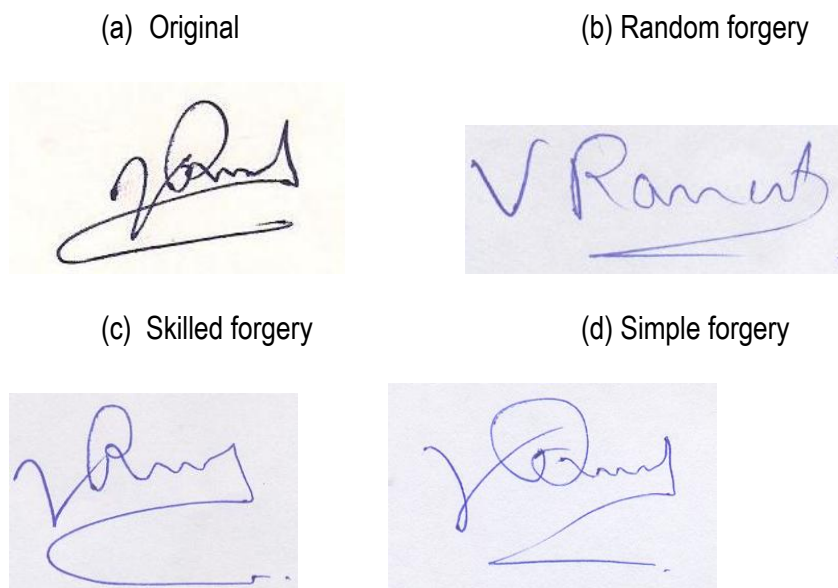
1.2. HOW IS IT DIFFERENT FROM CHARACTER RECOGNITION

Signature verification is so different with the character recognition, because signature is often unreadable, and it seems it is just an image with some particular curves that represent the writing style of the person. Signature is just a special case of handwriting and often is just a symbol. So it is wisdom and necessary to just deal with a signature as a complete image with special distribution of pixels and representing a particular writing style and not as a collection of letters and words

1.3. TYPES OF FORGERIES:

There are three different types of forgeries to take into account. First one is **random forgery** which is written by the person who doesn't know the shape of original signature. The second, called **simple forgery**, which is represented by a signature sample, written by the person who knows the shape of original signature without much practice. The last type is **skilled forgery**, represented by a suitable imitation of the genuine signature model. Each type of forgery requires different types of verification approach. Different types of forgeries and their variation from original signature are shown below in

Fig. 01: Original signature and different types of forgeries



The algorithm applied for off-line signature verification is typically a feature extraction method. In this method features are extracted and adjusted from the original signature(s). These extracted features are used to distinguish between original and forgery signatures. Two methods have been implemented for extraction of features, namely: -

(1) **CLUSTERING**

(2) **GEOMETRIC FEATURE EXTRACTION.**

CHAPTER 02:

METHODS OF OFF-LINE

SIGNATURE VERIFICATION

2.1. Template matching – warping based:

Warping is method to warp one of the curves onto other curve while attempting to preserve its original shape. Warping based matching methods produce the order of the coordinates and are able to match the coordinates that are placed in the same position in the two exterior curves that are formed by top and bottom parts of the template signature and test signature. To achieve similarity metric, the curve that is warped is represented in mass-spring system. The nodes in the graph will represent unit mass particles, whereas the springs are represented as edges. Disconnected curve parts form a separate graph. The first ring neighbors are the neighboring nodes which are connected to the node under study by a single edge. First-ring neighbors of a node are sorted by specifying a criterion such as angle, to provide structural springs which will be helpful for comparison between template and test signature. Classifier will be based on measuring intrinsic and extrinsic energy. Derivation of intrinsic energy is by structural curve constraints and extrinsic energy is from forces of attraction between the nodes of the template signature and test signature. In another novel method, vertical projection which is a one dimensional feature is used to perform non-linear matching on a rectangular grid by specifying least accumulated path cost function. In graph matching approach using deformable templates based on multi-resolution shape features, extremes along the contour for convexity and rotated versions the chain codes are extracted. The thin-plate spline mapping function is considered as an objective function for gradient-structural concavity to achieve region matching with contour directional chain codes. Feature extraction from a segmented signature can be done by Zernike moments. The global value for classification from all segments is derived from harmonic mean dissimilarity measure.

2.2. Hidden markov models:

This approach has ability to absorb both variability and the similarity between patterns in the problems that have an inherent temporality. This is based on statistically parameterized learning theory which is strictly causal. The training patterns form visible states. The production of sequence of transitions by corresponding probabilities are hidden. Signature length will depict the number of states with left-to-right topology. After running the simulations, learning probability will be stated. The topology only authorizes for transitions between each state to itself and to its immediate right-hand neighbours. A final or absorbing state is one which if entered, is never left. The classification (evaluation), decoding and learning problems are solved with forward backward algorithm, the Viterbi algorithm and Baum-Welch algorithm respectively. The threshold is defined by learning probability logarithm normalized by the observation number 'L', which is signature length. Forward algorithm is used to determine the verification probability. Test pattern is classified by the model that will have highest posterior probability.

2.3. Structural techniques

Structural features use modified direction and transition distance feature (MDF). MDF extracts the transition locations. In the boundary representation of an object transition can be from background to foreground pixels in the vertical and horizontal directions. The ratio between the position where a transition occurs and distance across the entire image in a particular direction will be termed as location of transition. The stroke direction of an object's boundary at position where a transition occurs will be termed as direction of transition. Centroid is an enhanced geometric feature. Classifier will be based on support vector machine [SVM] which reaches optimal separating plane by mapping nonlinearly the input vectors into a high dimensional feature space. Tri-surface and sixfold surface features will form some of the additional features. After normalization of all signatures to same height, the length of the signature forms length

feature. Euclidean distance with respect to average vector and test feature vector can be stated as another structural component. The average vector is average of training feature sequences of the genuine signature.

2.4. Feature –based techniques based on global features:

Global features are computed in subregions of the signature image. Fuzzy vault construction as biometric cryptosystem considers maxima and minima from upper and lower envelopes of the signature. Using the moving average method with span, smoothing envelopes can be obtained. Fuzzy vault input key is formed by set of quantized envelope values. Input size for the fuzzy vault will be set to 'n' bits. If the average matching points with respect to distance are 8 then it can be termed as good. The secret key and biometric template are stored in the system. The reconstruction will be possible only if a valid biometric trait is presented. According to work done by J.Francisco the envelope is extracted by first applying dilation followed by filling morphological operation. The research by J.Francisco explores how the image acquisition resolution affects the scenario of considering global parameters of signature, using smart card memory by a HMM classifier.

CHAPTER 03:

FUNDAMENTALS AND THEORY

3.1. CLUSTERING: -

Clustering is basically used in RBFNN (Radial Basis Function Neural Networks) for auto up gradation or adjustment of centres of the network to get an efficient output. We applied clustering for extraction of features from the original signature image. The obtained adjusted centres are considered as feature points of the signature image. In this method of feature extraction, all the co-ordinates of the signature are taken as input of the network (feature adjustment network). Equally spaced feature points are taken at random. The random feature points are adjusted using the equation: -

$$t_{k+1}(n) = t_k(n) + \text{eta} * (x(i) - t_k(n))$$

where,

$t_{k+1}(n)$ – $(k+1)^{\text{th}}$ iteration number

t – Centre or feature point

k – Iteration number

n – Centre number.

eta – centre adjustment parameter (step size)

$x(i)$ – i^{th} input

This actually tries to move the centre closest to the input towards the input according to a pre-defined step-size “eta”. After a few iterations (100 in our project) the centres adjust themselves completely and can be considered as the feature points of the signature image. These feature points are mapped on to the signature image to be tested. A threshold is decided as the rule to distinguish between original and forgery signatures. Based on that specific threshold it is decided whether the signature is a forgery or the original one.

3.2. GEOMETRIC FEATURE EXTRACTION:

In this method of feature extraction, the signature image is divided into several sub-images or blocks, like blocks containing equal number of pixels (signature pixels generally black pixels), equi-spaced-equal-sized blocks etc. Then the centroid of each of these blocks or sub-images acts as a feature points of the image. Rest of the process is same as that described in the previous section (CLUSTERING).

3.3. IMAGE BINARIZATION:

The input image is taken and converted into gray-scale image which is further pre-processed i.e. it is binarized. The binary image of the signature contains only 0's and 1's. Where 0's represents the signature boundary and 1's represents the blank white area or the background region. This is done by specifying a specific threshold, above which every gray value is 1 and below which every gray value is 0.

CHAPTER 04:

APPLIED ALGORITHM

4.1. APPLIED ALGORITHM: -

- TAKE INPUT THE ORIGINAL SIGNATURE IMAGE.
- CONVERT IT INTO BINARY IMAGE BY TAKING CARE OF THE NOISE PRESENT IN THE IMAGE.
- SKELTENIZE THE IMAGE.
- FIND THE LOCATIONS OF ALL THE BLACK PIXELS (SIGNATURE PIXELS) FROM THAT IMAGE. (BACK GROUND BEING WHITE).
- ASSIGN EQUALLY SPACED CENTRES TO THE IMAGE SIGNATURE.
- BY TAKING THE X-Y CO-ORDINATES OF THE BLACK PIXELS OF THE IMAGE AS INPUT ADJUST THE CENTRES USING THE EQUATION DESCRIBED ABOVE.
- ADJUST THEM FOR A FEW TRAINING SIGNATURE IMAGES (OF A SAME PERSON).
- TAKE THE FINALLY ADJUSTED CENTRES AS THE FEATURE POINTS OF THE SIGNATURE.
- CREATE A DISTANCE MATRIX FROM THE FEATURE POINTS (WHICH CONTAINS DISTANCE OF EACH FEATURE POINT FROM EVERY OTHER FEATURE POINT).
- SCALE THE DISTANCES BY A FACTOR OF 100. (THIS MAKES IT ROTATION AND SCALE INVARIANT)
- TAKE INPUT THE SIGNATURE IMAGE TO BE TESTED.
- CONVERT IT INTO BINARY IMAGE.
- EXTRACT ITS FEATURES ACCORDING TO THE STEPS DISCUSSED ABOVE.
- NOW CREATE A DISTANCE MARIX IN THE SAME WAY.
- SCALE THESE DISTANCES BY A FACTOR OF 100.
- MAP THIS DISTANCE MARIX WITH THE FEATURE POINT DISTANCE MATRIX.

- NUMBER OF ERRORS LESS THAN 5.0 SHOULD BE ABOVE 18 AND TOTAL ERROR IS LESS THAN 155
- IF THE TEST SIGNATURE MATCHES BOTH THE ABOVE MENTIONED CRITERIA THEN THE SIGNATURE IS DECLARED ORIGINAL ELSE IT IS DECLARED FORGERY.

CHAPTER 05:

SOURCE CODE

MATLAB SOURCE CODE FOR COMPARING DISTANCE MATRIX WITH THE ROWS OF THE DISTANCE MATRIX: -

```
clc;
close all;
clear all;

eta = 0.005; %center adjustment step size.....

x1 = rgb2gray(imread('D:\Study\Project 2008-
09\signature_database\p1_s8.jpg'));
%x1 =
rgb2gray(imread('C:\Users\Kaushal\Pictures\sig_name009.jpg'));
x1 = im2bw(x1);
x1 = 1-x1;
figure;
imshow(x1);
x = bwmorph(x1,'skel',inf);
x = 1-x;
figure
imshow(x);

[m n] = size(x);

for i = 1:1:m
    if(min(x(i,:)) == 1)
        continue;
    else
        x_min = i;
        break;
    end
end

for i = m:-1:1
    if(min(x(i,:) == 1))
        continue;
    else
        x_max = i;
        break;
    end
end
```

```

for i = 1:1:n
    if(min(x(:,i) == 1))
        continue;
    else
        y_min = i;
        break;
    end
end

for i = n:-1:1
    if(min(x(:,i) == 1))
        continue;
    else
        y_max = i;
        break;
    end
end

y = x(x_min:x_max,y_min:y_max);
figure;
imshow(y);
y = rot90(y);
y = rot90(y);
y = rot90(y);

[m n] = size(y);

sig_ind = 0;

for i = 1:1:m
    for j = 1:1:n

        if (y(i,j)<1)
            sig_ind = sig_ind+1;
            xdx(sig_ind,1) = i;
            ydx(sig_ind,1) = j;
        end
    end
end

sig_ind = 0;

%[xdx, ydx] = find(y<1);
len_pix = length(xdx);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CLUSTERING STARTS HERE.....
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_inc = round((m - mod(m,8))/8);
y_inc = round((n - mod(n,7))/7);

ind = 0;
for i = x_inc:x_inc:(m-x_inc)
    ind = ind + 1;
    cx(ind) = i;
end
ind = 0;
for i = y_inc:y_inc:(n-y_inc)
    ind = ind + 1;
    cy(ind) = i;
end
ind = 0;
for i = 1:1:(length(cx))
    for j = 1:1:(length(cy))
        ind = ind+1;
        t(1,ind) = cx(i);
        t(2,ind) = cy(j);
    end
end
ind = 0;
figure;
plot(xdx,ydx, '.', 'color', 'c');
hold on;
plot(t(1,:),t(2,:), 'p', 'color', 'r');
xlim([0 m]);
ylim([0 n]);

t_len = length(t); %total number of centres.....

for k=1:1:100

    for i = 1:1:len_pix
        for j = 1:1:t_len
            d(j) = sqrt(((t(1,j)-xdx(i,1))^2) + ((t(2,j)-
ydx(i,1))^2));
        end

        d_min = min(d);
        ind = find(d<=d_min);
        ind = ind(1,1);
    end
end

```

```

temp_input = [xdx(i,1) ydx(i,1)]';
t(:,ind) = t(:,ind) + eta*(temp_input - t(:,ind));

end

end

figure;
plot(xdx,ydx, '.', 'color', 'c');
hold on;
plot(t(1,:),t(2,:), 'o', 'color', 'r');
xlim([0 m]);
ylim([0 n]);

for i = 1:1:t_len
    for j = 1:1:t_len

        d_center = sqrt(((t(1,i)-t(1,j))^2) + ((t(1,i)-
t(1,j))^2));

        mat_dist(i,j) = d_center;
    end
end

for i = 1:1:t_len
    d_max = max(mat_dist(i,:));
    mat_dist(i,:) = (mat_dist(i,:)/d_max)*100;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SIGNATURE IMAGE TO BE TESTED.....
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

eta = 0.005; %center adjustment step size.....

```

```

x1 = rgb2gray(imread('D:\Study\Project 2008-
09\signature_database\p1_s3.jpg'));
%x1 =
rgb2gray(imread('C:\Users\Kaushal\Pictures\sig_name010.jpg'));
x1 = im2bw(x1);
x1 = 1-x1;
x = bwmorph(x1,'skel',inf);
x = 1-x;

```

```

[m n] = size(x);

```

```

for i = 1:1:m
    if(min(x(i,:) == 1))
        continue;
    else
        x_min = i;
        break;
    end
end

```

```

for i = m:-1:1
    if(min(x(i,:) == 1))
        continue;
    else
        x_max = i;
        break;
    end
end

```

```

for i = 1:1:n
    if(min(x(:,i) == 1))
        continue;
    else
        y_min = i;
        break;
    end
end

```

```

for i = n:-1:1
    if(min(x(:,i) == 1))
        continue;
    else
        y_max = i;
        break;
    end
end

```

```

end

y = x(x_min:x_max,y_min:y_max);
y = rot90(y);
y = rot90(y);
y = rot90(y);

[m n] = size(y);

sig_ind = 0;

for i = 1:1:m
    for j = 1:1:n

        if (y(i,j)<1)
            sig_ind = sig_ind+1;
            xdx(sig_ind,1) = i;
            ydx(sig_ind,1) = j;
        end
    end
end

sig_ind =0;
%[xdx, ydx] = find(y<1);
len_pix = length(xdx);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CLUSTERING STARTS HERE.....
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_inc = round((m - mod(m,8))/8);
y_inc = round((n - mod(n,7))/7);

ind = 0;
for i = x_inc:x_inc:(m-x_inc)
    ind = ind + 1;
    cx(ind) = i;
end
ind = 0;
for i = y_inc:y_inc:(n-y_inc)
    ind = ind + 1;
    cy(ind) = i;
end

```



```

ind =0;
for i = 1:1:(length(cx))
    for j = 1:1:(length(cy))
        ind = ind+1;
        t(1,ind) = cx(i);
        t(2,ind) = cy(j);
    end
end
ind = 0;

t_len = length(t); %total number of centres.....

for k=1:1:100

    for i = 1:1:len_pix
        for j = 1:1:t_len
            d(j) = sqrt(((t(1,j)-xdx(i,1))^2) + ((t(2,j)-
ydx(i,1))^2));
        end

        d_min = min(d);
        ind = find(d<=d_min);
        ind = ind(1,1);
        temp_input = [xdx(i,1) ydx(i,1)]';
        t(:,ind) = t(:,ind) + eta*(temp_input - t(:,ind));

    end

end

figure;
plot(xdx,ydx, '.', 'color', 'c');
hold on;
plot(t(1,:),t(2,:), 'o', 'color', 'r');
xlim([0 m]);
ylim([0 n]);

% for i = 1:1:t_len
%     d_center = sqrt(((t(1,i)-t(1,1))^2) + ((t(1,i)-
t(1,1))^2));
%     vec_d(1,i) = d_center;
% end

for i = 1:1:t_len
    for j = 1:1:t_len

```

```

        d_cntr = sqrt(((t(1,i)-t(1,j))^2) + ((t(1,i)-
t(1,j))^2));

        mat_test(i,j) = d_cntr;
    end
end

for i = 1:1:t_len
    d_max = max(mat_test(i,:));
    mat_test(i,:) = (mat_test(i,+)/d_max)*100;
end

% max_vec = max(vec_d);
% vec_d = (vec_d/max_vec)*100;
% vec_d2 = vec_d;

% for i = 1:1:t_len
% %     for j = 1:1:t_len
% %
% %         e(i,:) = abs(mat_dist(i,:) - vec_d2);
% %
% %         vt = vec_d2(1,t_len);
% %         vec_d2(1,2:t_len) = vec_d2(1:(t_len-1));
% %         vec_d2(1,1) = vt;
% %         vec_d2
% end

e = abs(mat_dist - mat_test);
e_sum = (sum(e(3,:))/t_len); % a random index value

% et2 = (sum(e(10,:))/t_len);
eind = 10;

for i = 1:1:t_len
    et(1,i) = (sum(e(i,:))/t_len);
    if (et(1,i)<e_sum)
        e_sum = et(1,i);
        eind =i;
    end
end

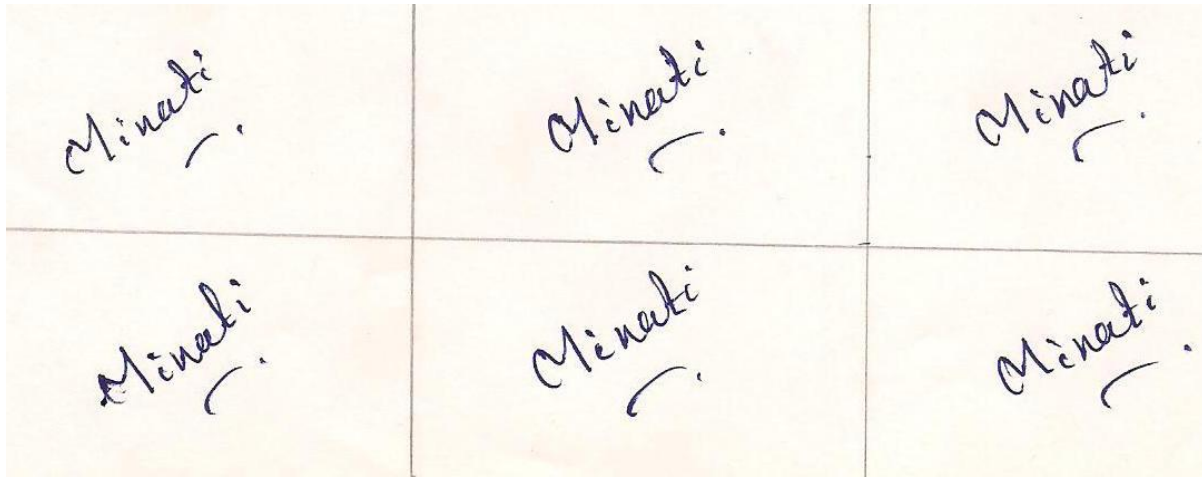
fr = e(eind,:)

```

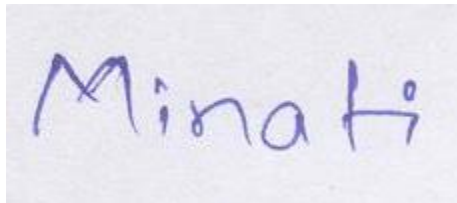
CHAPTER 06:

SIMULATION RESULTS

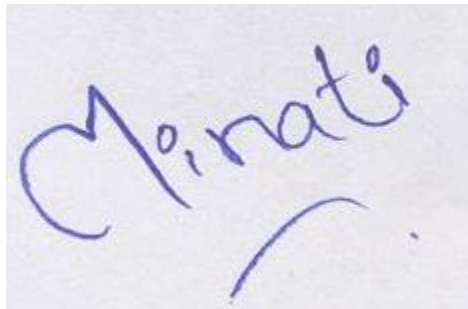
Training signatures for person # 01



Test Signatures for person # 01



Random forgery

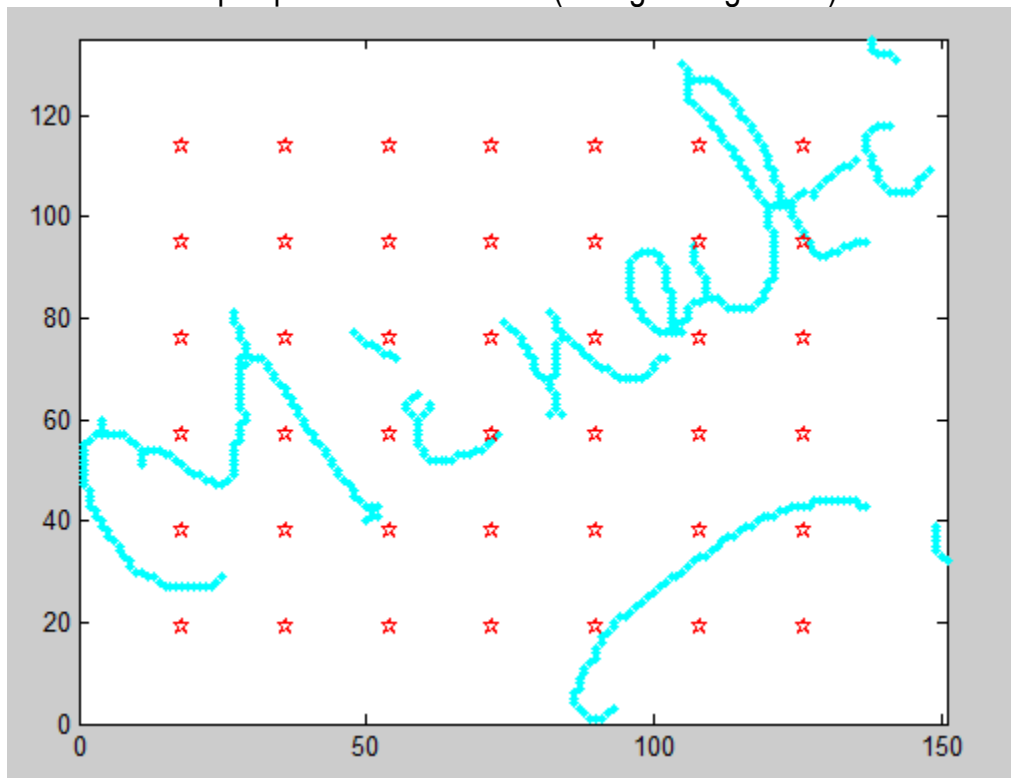


Skilled forgery

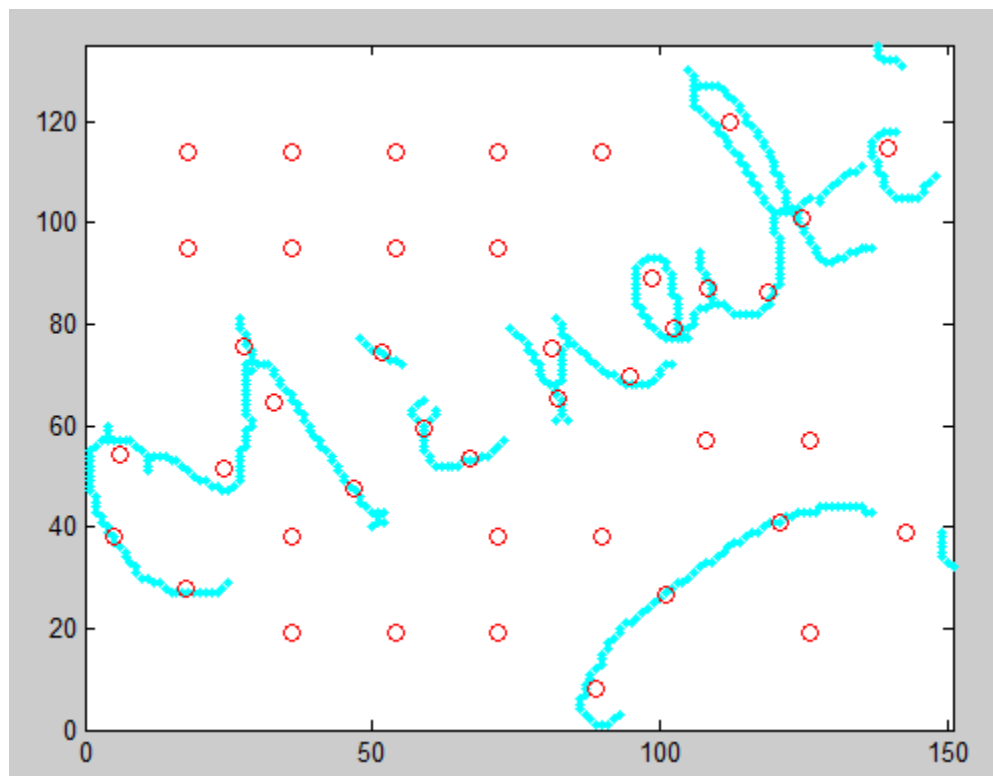


Simple forgery

Equi-spaced initial centres (in original signature)

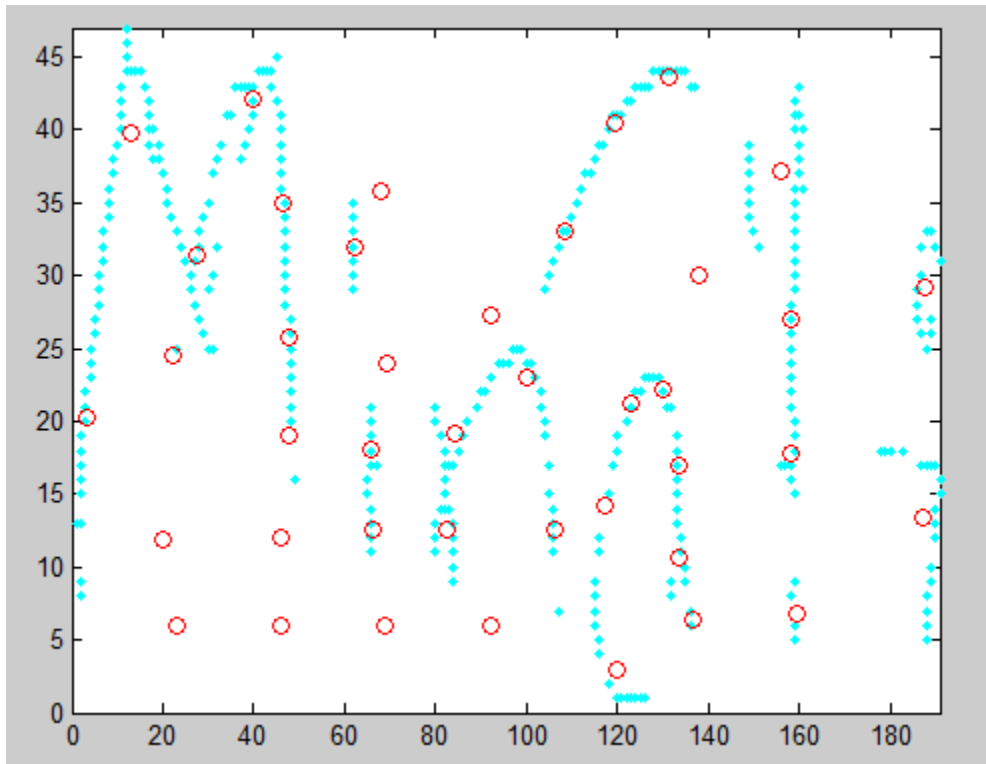


Trained and adjusted centres from training data



FOR RANDOM FORGERY

Clustered centres from test signature image (random forgery)



Minimum error: -

fr = 3.8855 5.6970 7.3937 13.5165 0.9046 11.2706 1.7075 1.7075 7.6664
0 2.0949 2.6911 4.0661

0.5201 10.8420 1.7025 8.8877 4.6761 4.0268 10.9060 5.1302 6.6276
3.8736 18.8493 0.2915 10.1321

4.3608 2.9414 16.3695 0.7410 1.4523 18.8276 7.1906 4.8039 4.1288
12.2374 4.9268 21.2151 14.8094

0.8967 16.3607 20.2249

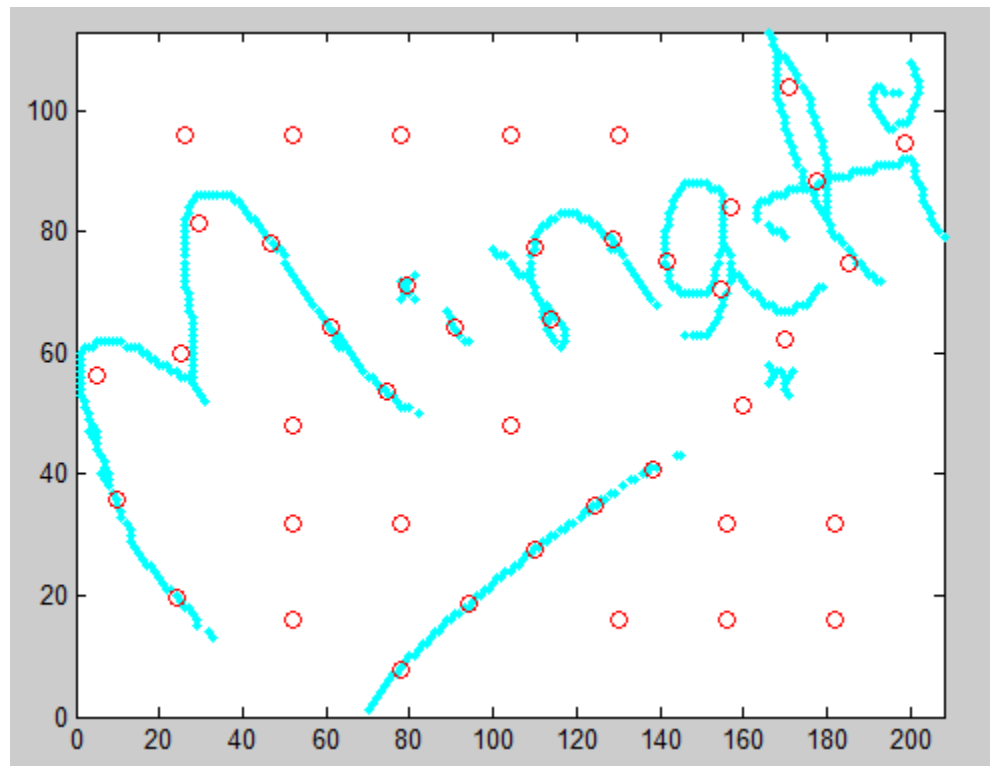
Number of errors below 5.0 = 22

Total error = 300.5536

Signature is declared forgery

FOR SIMPLE FORGERY

Clustered centres from test signature image (simple forgery)



Minimum error: -

fr = 0 0.2972 0.4655 0.6715 0.8546 1.2327 1.2518 1.3344 1.3413 1.3413
1.3413 1.3757 1.4309

1.5205 1.6101 1.7893 1.9674 2.1659 2.3028 2.4495 3.1579 3.4419 3.7092
3.9620 4.0703 4.5192

4.5293 4.6734 4.9873 5.1406 5.4762 6.2349 6.3084 6.3171 6.6765 6.9446
7.8854 8.3794 8.8009

9.8138 10.0533 11.3929

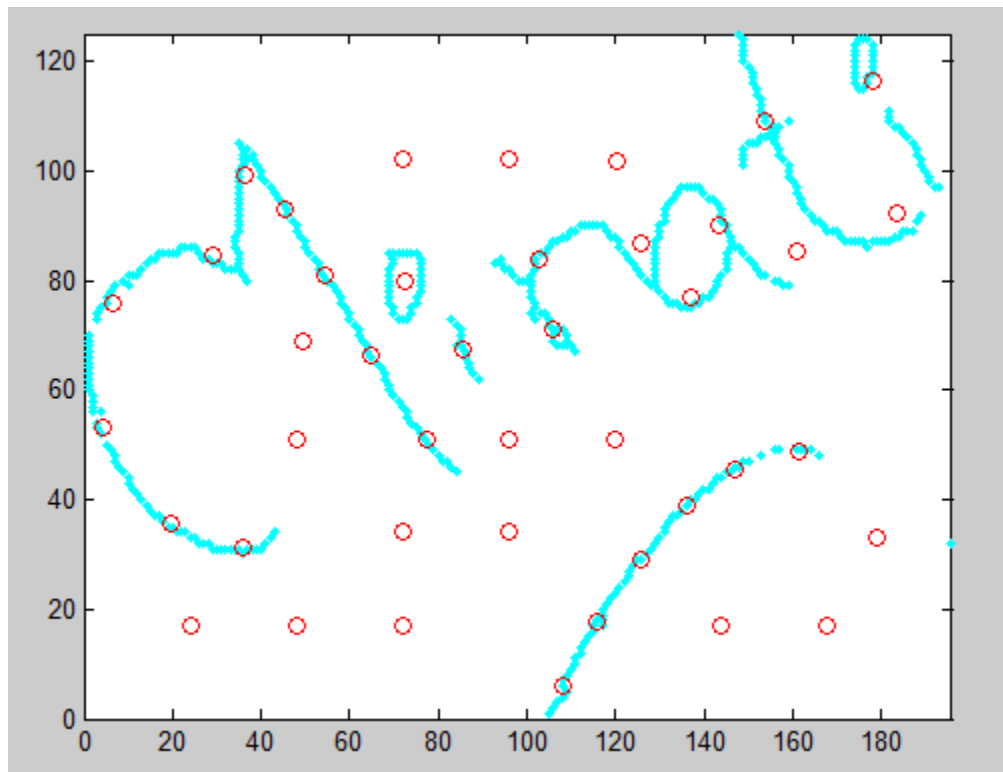
Number of errors below 5.0 = 29

Total error = 163.2185

Signature is declared forgery

FOR SKILLED FORGERY

Clustered centres from test signature image (skilled forgery)



Minimum error: -

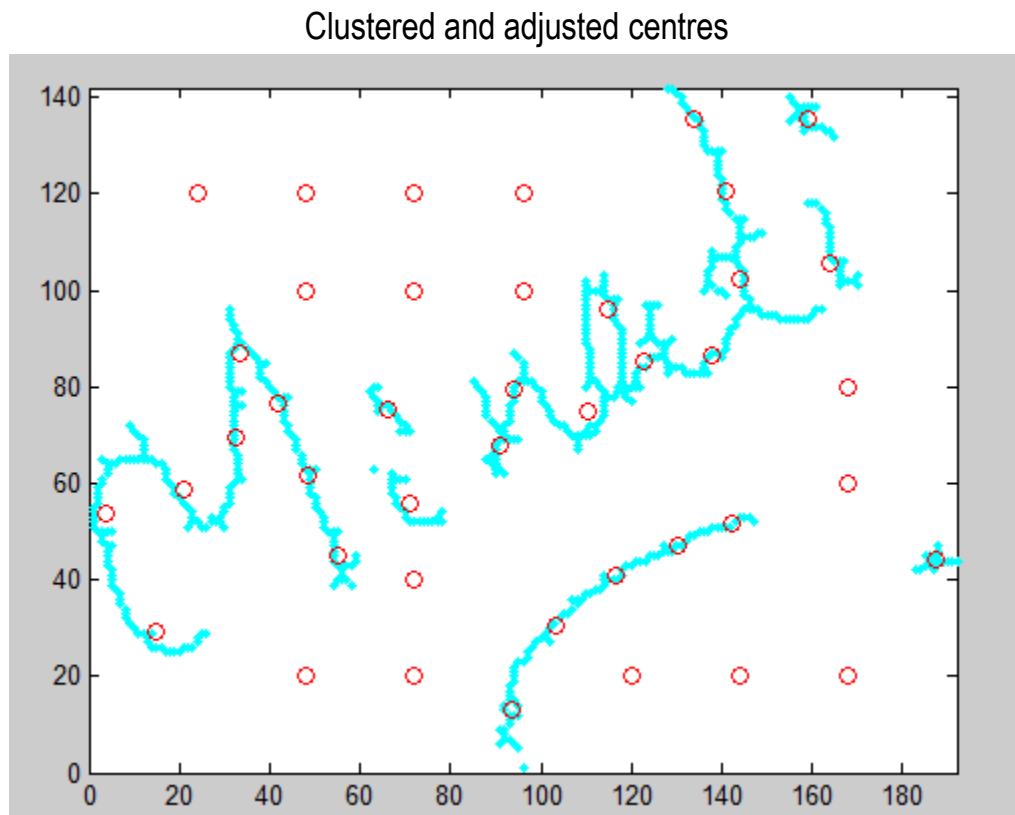
fr = 3.0108 4.1482 6.4594 6.8658 0 4.8014 2.1986 10.0823 7.2150
1.1641 1.8404 3.7733 1.0921
4.5760 1.7728 3.7271 0.8294 1.0921 7.9786 0.0145 3.9262 14.1003
4.2472 0.0145 0.7810 4.9612
7.1649 11.8723 2.0388 1.2627 7.8950 13.0756 4.0552 1.9880 1.5585
5.1471 3.3342 2.9932 0.8355
4.7206 14.4203 1.2647

Number of errors below 5.0 = 30

Total error = 184.2990

Signature is declared forgery

FOR ONE OF THE TRAINING IMAGES (ONE OF THE ORIGINAL SIGNATURES)



Minimum error: -

fr = 3.7418 0 9.4703 1.2379 7.1888 1.5718 1.3098 5.6877 11.2710

0.1389 1.3098 1.3098 1.0479

6.8946 3.7598 0.4436 1.0479 1.0479 0.7484 5.1400 1.7155 8.0819 0.7859

0.7859 1.2960 1.5661

0.9703 1.8760 9.8269 8.8325 6.1079 18.8242 0.8255 0.9222 0.1938

4.9243 0 2.1612 0

6.0673 1.5684 5.9786

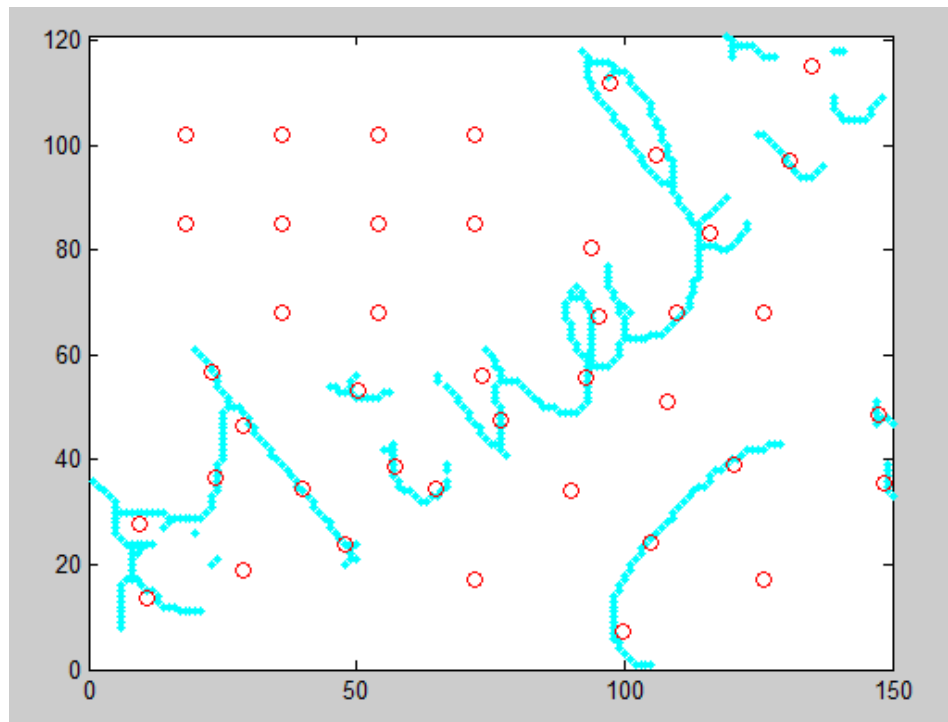
Number of errors below 5.0 = 29

Total error = 147.6781

Signature is declared as original

FOR ONE OF THE TRAINING IMAGES (ORIGINAL SIGNATURE 4th in database)

Clustered and adjusted centres



Minimum error: -

fr = 5.1166 3.9419 5.1456 3.9355 0 0 6.1755 2.2437 3.2561 1.9293
0.6324 0.6324 5.9604

6.8410 8.1508 0.6811 1.2649 1.2649 1.8973 7.4401 5.7559 8.3913 1.8973
1.8973 5.6423 2.5297

5.7034 2.5698 6.5266 3.0241 0.0738 4.0762 3.1622 2.3609 2.6659 7.9852
3.7946 0 12.5735

2.0873 0.9783 8.1060

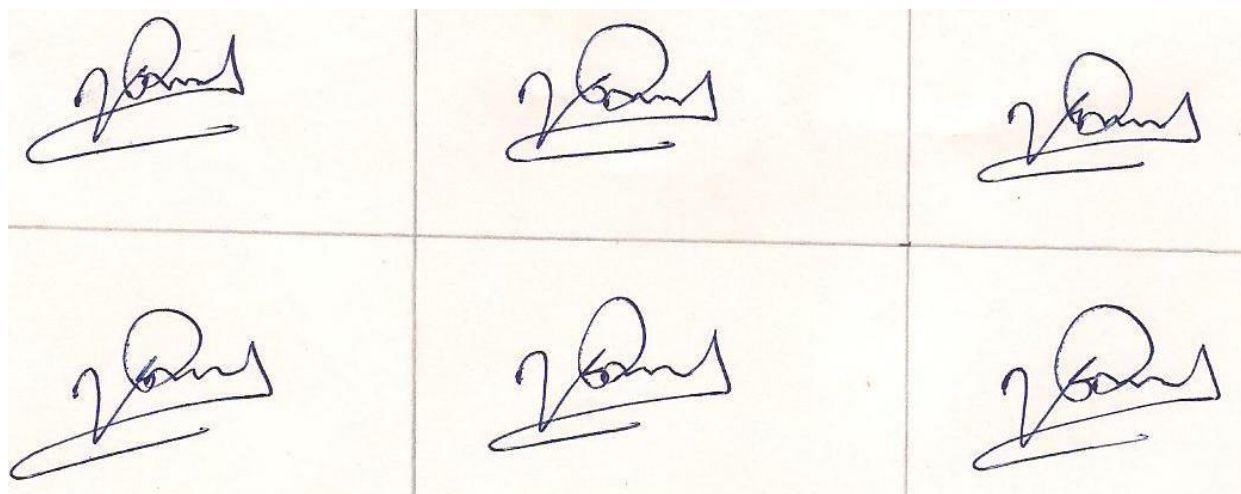
Number of errors below 5.0 = 27

Total error = 158.3110

Signature is declared forgery

(Original signature is declared forgery because of interpersonal variations)

Training signatures for person # 02



Test signatures for person # 02



Random forgery

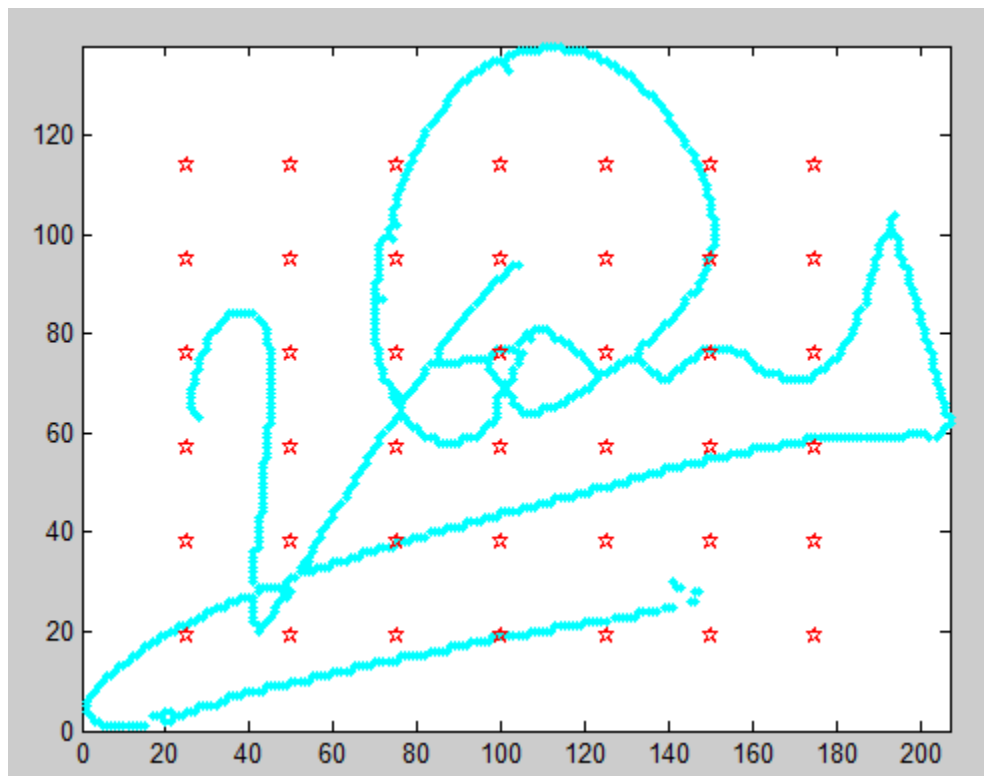


Skilled forgery

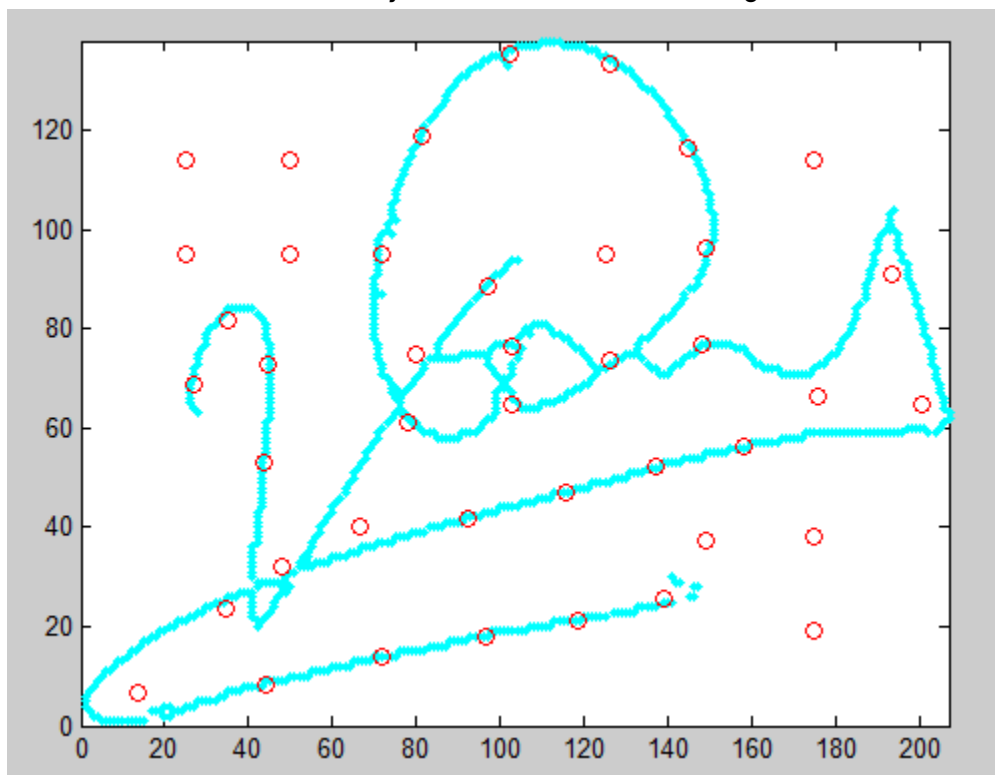


Simple forgery

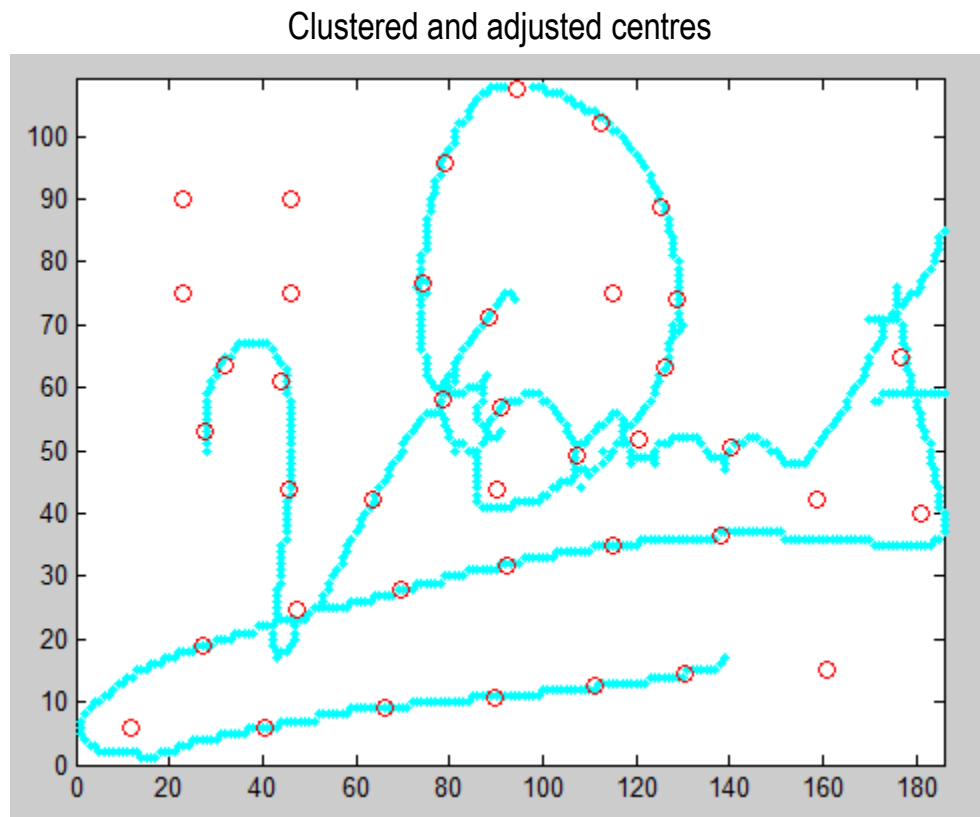
Equi-spaced initial centres (in original signature)



Trained and adjusted centres from training data



FOR ONE OF THE TRAINING IMAGES (ONE OF THE ORIGINAL SIGNATURES)



Minimum error:

fr = 0 2.2564 2.2871 0.3910 0.5184 0.5184 0.2279 2.4898 3.8539 2.2487
0.5229 0.5229 0.6552

5.7881 4.2187 3.6236 5.4377 3.0494 1.2047 4.9625 1.9778 1.7416
0.0813 0.8590 1.9769 5.8465

10.7490 3.4669 0.5362 1.6201 2.3103 1.0901 2.4764 5.4105 4.4357
4.2153 0.5451 7.5472 6.6873

4.9956 0 0.2355

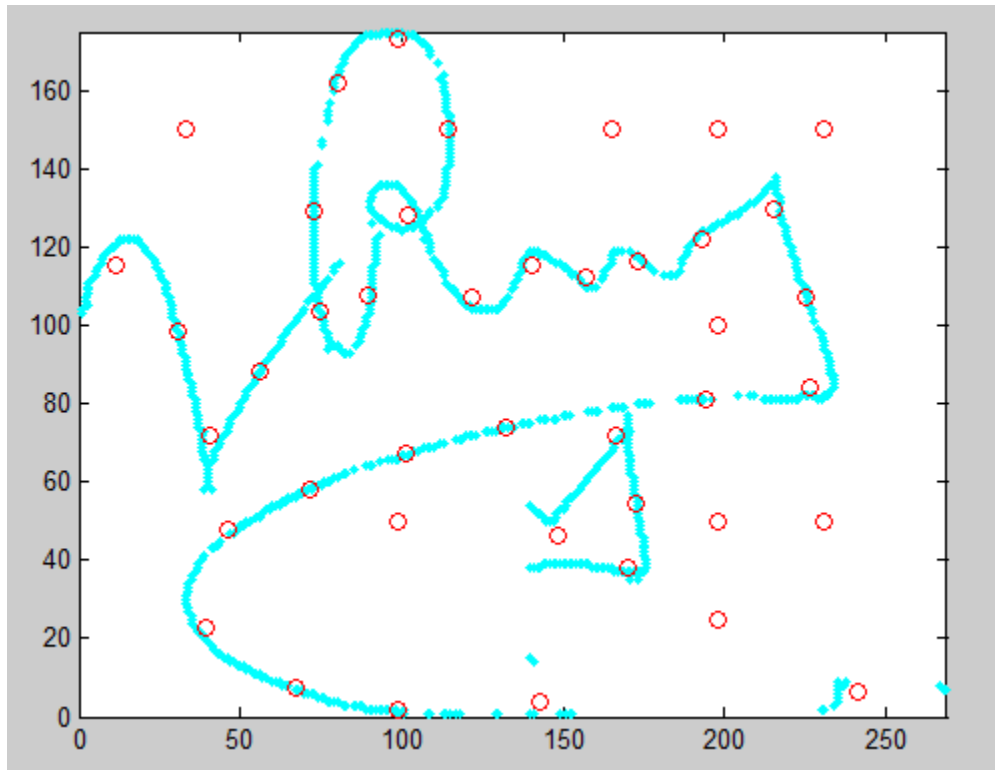
Number of errors below 5.0 = 33

Total error = 113.5817

Signature is declared as original.

FOR SKILLED FORGERY

Clustered centres from test signature image (skilled forgery)



Minimum error: -

fr = 1.2713 10.6996 2.9555 11.3272 32.0911 14.1139 3.9040 1.4528 7.6780
0 4.2892 2.2582 6.0076

6.6895 4.2935 13.1297 2.1346 9.7727 2.4123 6.0117 1.4779 9.4869
7.0281 15.2886 2.0508 3.8677

5.9981 9.6593 4.5004 1.2167 2.9162 2.5976 0.1763 6.7757 1.2494 5.4686
12.8974 6.3317 6.3083

7.4185 3.4775 6.5207

Number of errors below 5.0 = 20

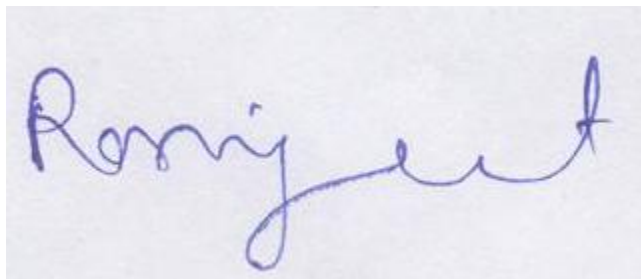
Total error = 265.2052

Signature is declared forgery

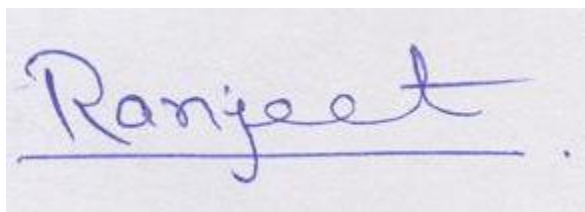
Training signatures for person # 03



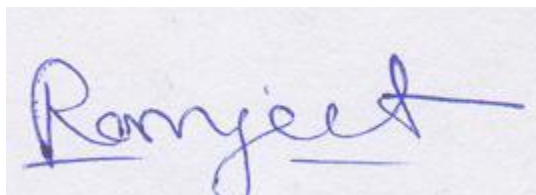
Test Signatures for person # 03



Random forgery

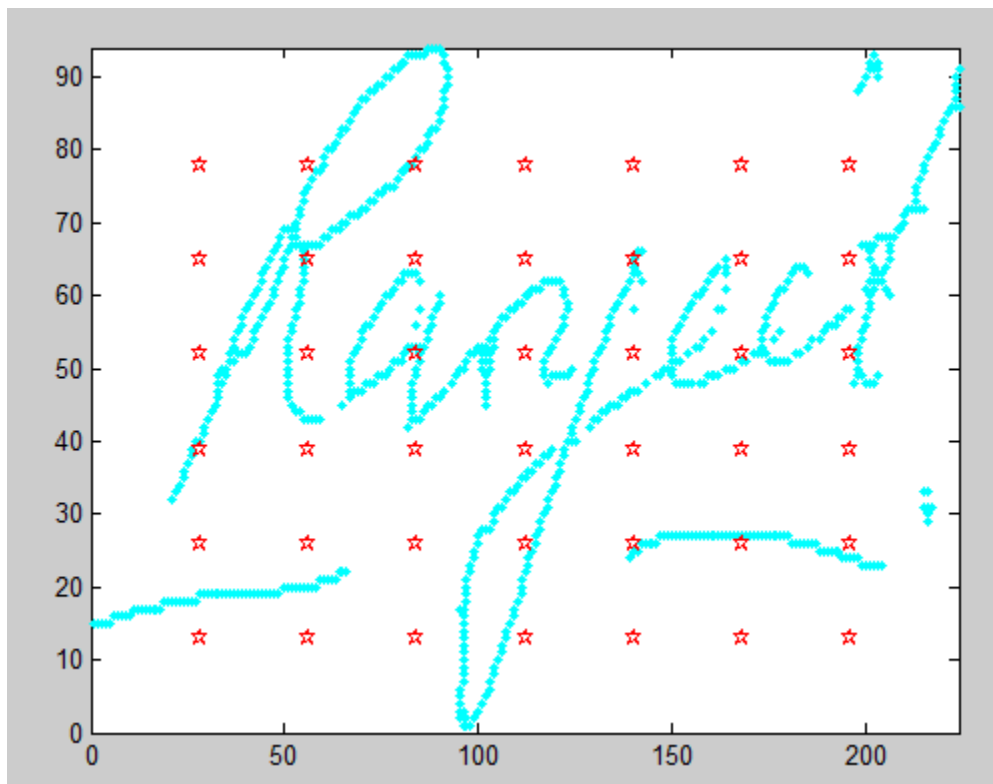


Simple forgery

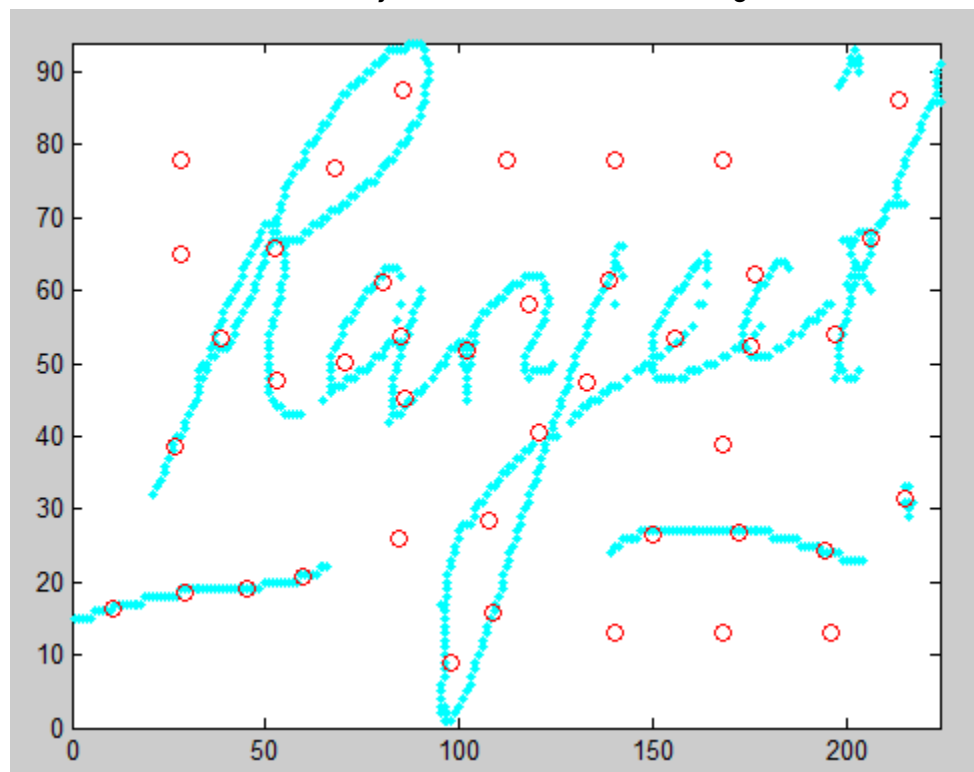


Skilled forgery

Equi-spaced initial centres (original image)

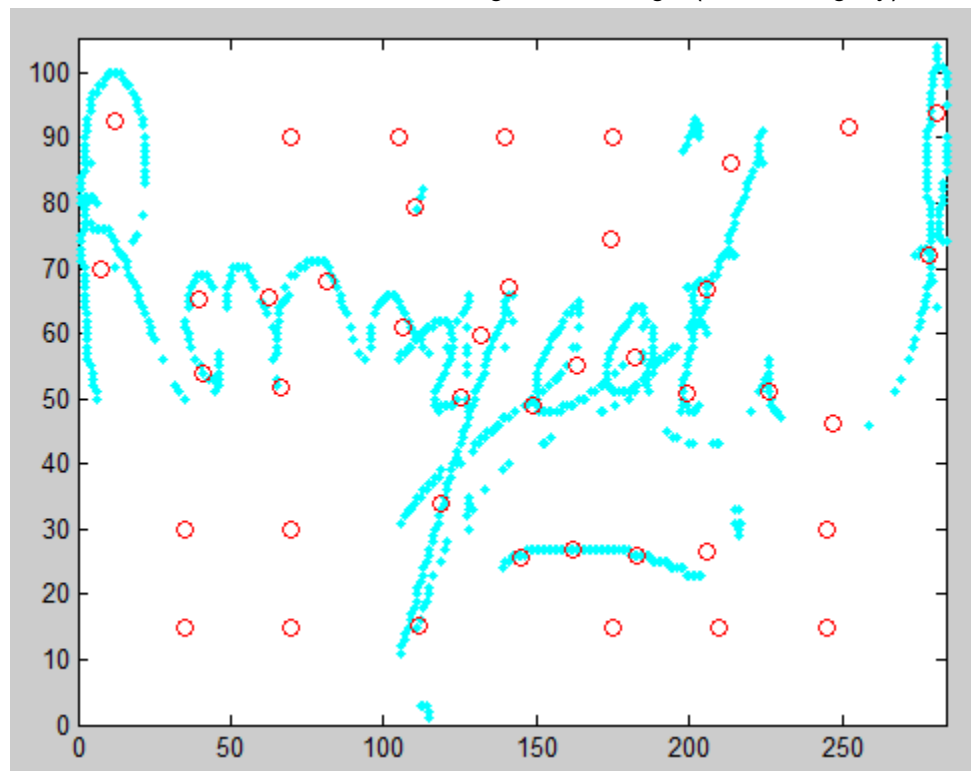


Trained and adjusted centres from training data



FOR SKILLED FORGERY

Clustered centres from test signature image (skilled forgery)



Minimum error: -

fr = 0 0.0158 0.0158 0.3177 0.4490 0.4702 0.5017 0.5029 0.6053 0.9877
1.0172 1.0678 1.1602

1.4158 1.4185 1.8628 2.0148 2.3819 2.7038 3.7254 4.5519 4.5560 4.9494
5.1936 5.2850 5.4595

6.1721 6.6302 6.6419 7.3172 7.5100 7.8237 8.1272 8.4285 8.6304 9.0417
9.4005 9.6431 10.5079

11.5330 12.1489 12.6388

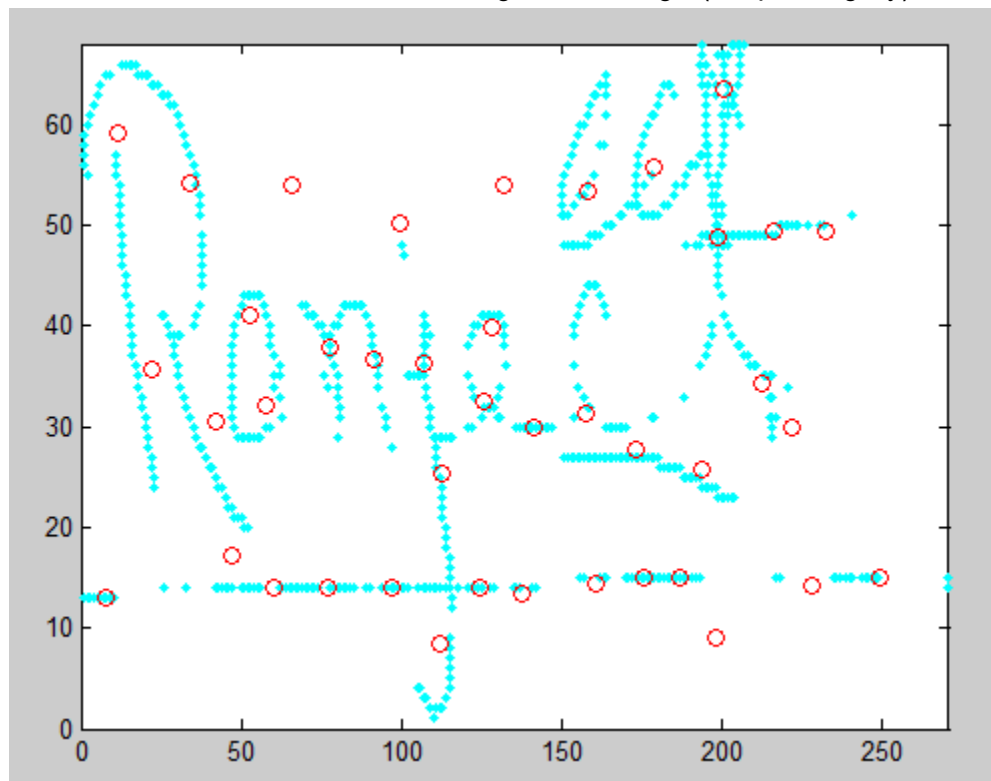
Number of errors below 5.0 = 23

Total error = 194.8247

Signature is declared forgery

FOR SIMPLE FORGERY

Clustered centres from test signature image (simple forgery)



Minimum error: -

fr = 0 7.8019 6.8666 8.3643 2.4356 7.6028 12.6866 2.6237 0.1829 0.5150
2.2117 4.4473 0.3482

0.8918 6.9572 4.9038 0.1674 1.5184 5.9753 0.5615 1.3677 4.3469 3.1892
1.6131 0.4024 1.0237

8.9874 10.2584 0.9483 7.7412 1.3446 5.8094 0.5009 2.2845 1.7963
8.4169 0 8.4765 7.4287

6.5701 3.5842 4.0157

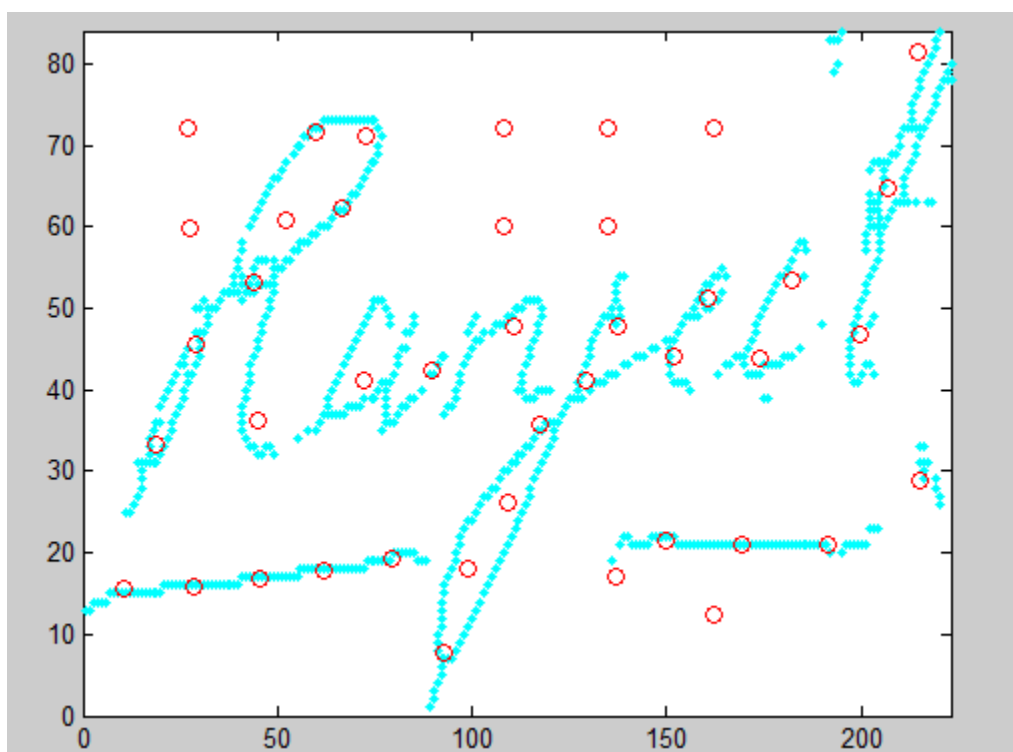
Number of errors below 5.0 = 27

Total error = 167.1682

Signature is declared forgery

FOR ONE OF THE TRAINING IMAGES (ONE OF THE ORIGINAL SIGNATURES)

Clustered and adjusted centres



Minimum error: -

fr = 0 0.5988 3.5158 3.8015 0.2802 0.1782 1.6058 3.1800 2.5456 11.4727
1.5829 1.7992 0.7826

0.4901 5.1283 3.5667 4.6704 3.5286 1.0941 4.6298 2.6753 8.4542 0.9380
1.8883 3.4101 5.4798

2.8038 3.8614 3.1152 2.4584 2.9639 4.8131 9.2294 5.8477 1.6808 3.0285
4.7582 2.0826 0

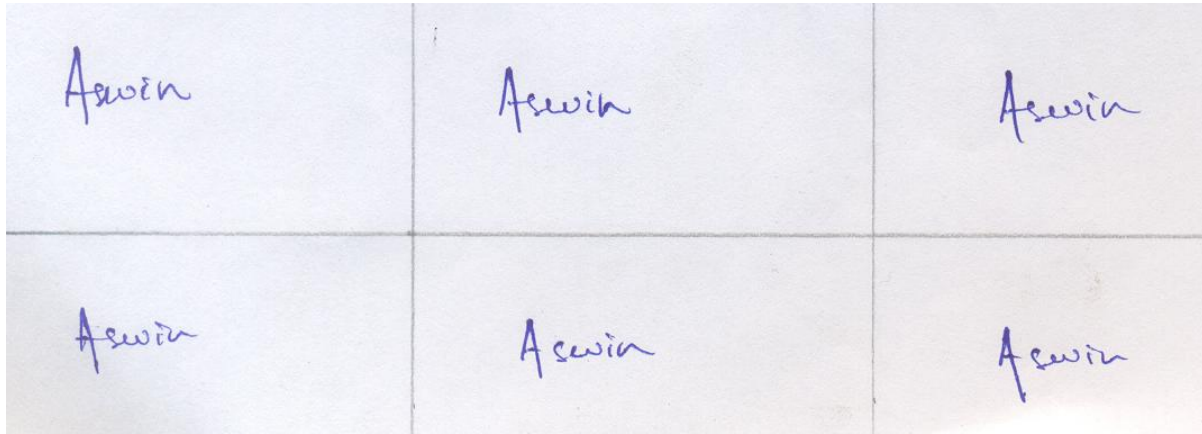
0.5333 0.7050 7.2019

Number of errors below 5.0 = 35

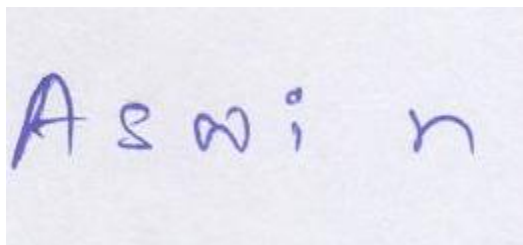
Total error = 132.3800

Signature is declared as original

Training signatures for person # 04



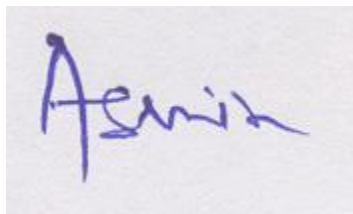
Test Signatures for person # 04



Random forgery

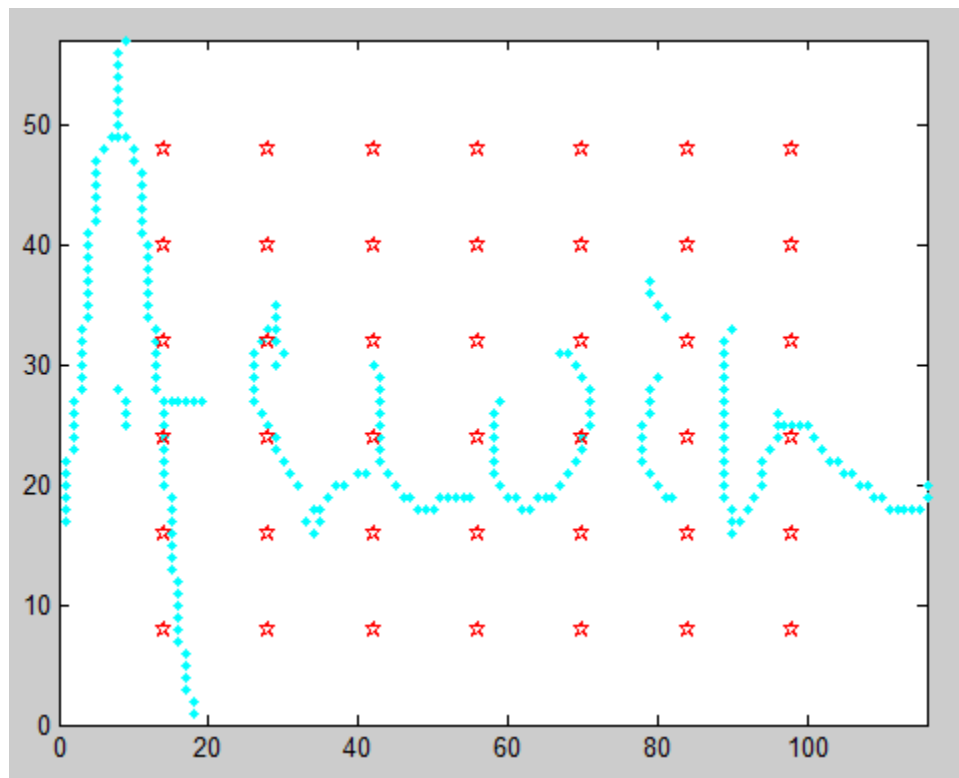


Simple forgery

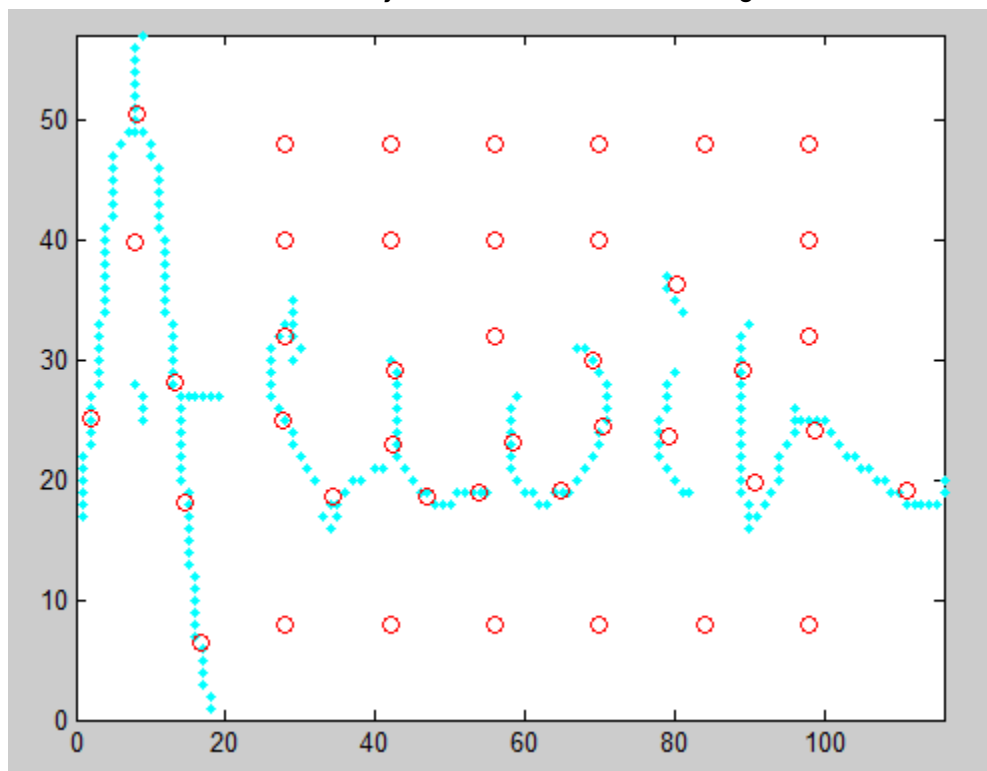


Skilled forgery

Equi-spaced initial centres (original image)

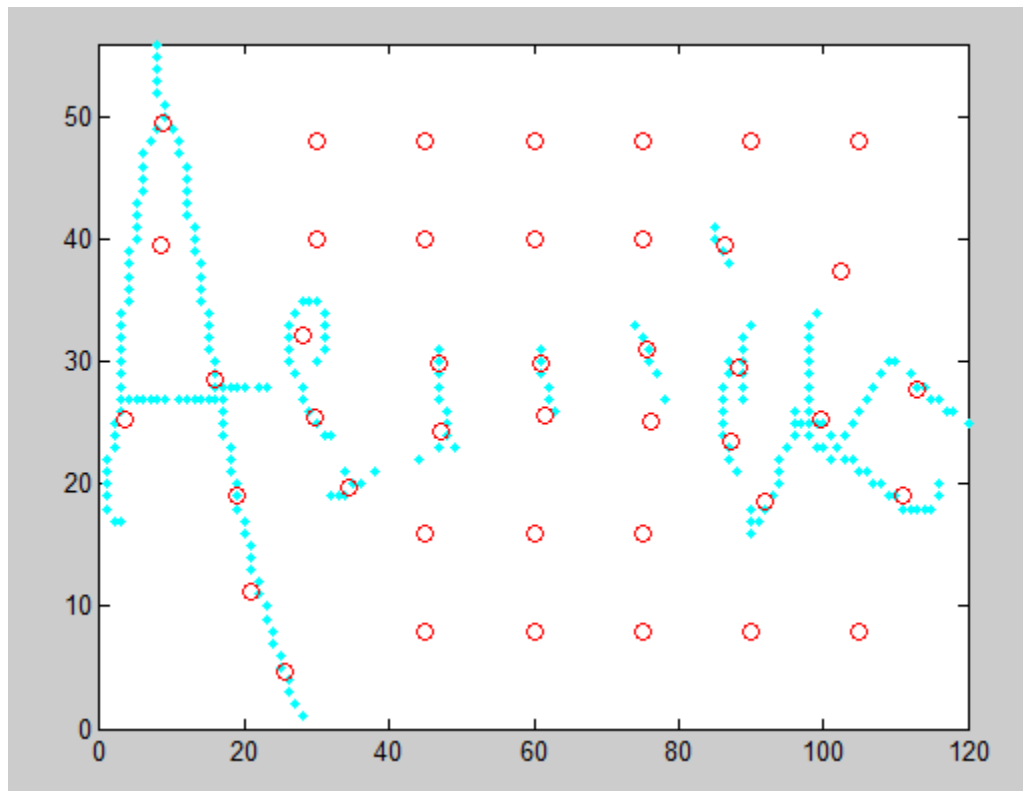


Trained and adjusted centres from training data



FOR ONE OF THE TRAINING IMAGES (ONE OF THE ORIGINAL SIGNATURES)

Clustered and adjusted centres



Minimum error: -

```
fr = 2.0833  2.3206    0  0.6758  1.0903  0.9977  5.1735  3.1815  0.7479
2.6480  0.8397  0.8397  0.6718

5.8641  1.0245  0.3849  0.6718  0.6718  0.5038  1.8525  1.4033  0.5620  0.5038
0.5038  0.3359  5.1834

0.5096  1.1601  0.3359  0.3359  0.1679  5.3222  1.8904  7.1644  0.0326  0.1679
0  7.6136  4.4184

7.8049  2.5049    0
```

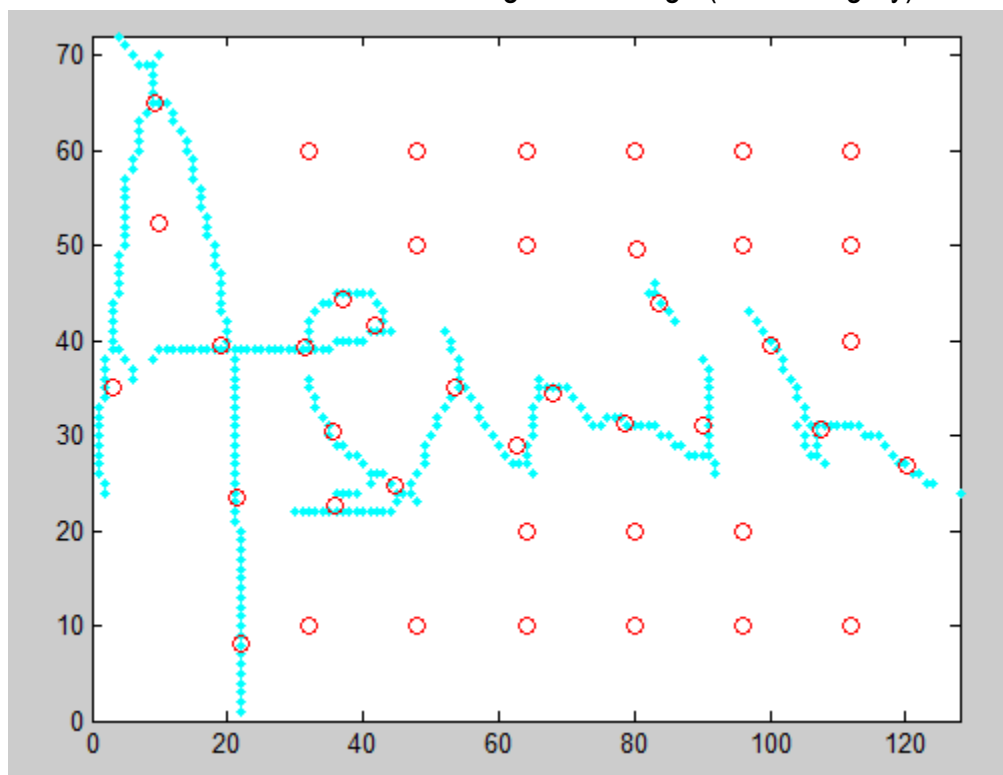
Number of errors below 5.0 = 35

Total error = 80.1642

Signature is declared as original

FOR SKILLED FORGERY

Clustered centres from test signature image (skilled forgery)



Minimum error: -

fr = 2.1520 3.6616 0 3.0075 0.1462 0.5738 0.4727 3.7484 3.1261
1.0103 4.0005 0.4727 0.3781

8.5224 4.2218 6.9091 0.3781 0.3781 0.2836 2.0728 3.9026 3.4511 0.2836
0.2836 0.1891 5.3303

1.7517 4.0973 0.1230 0.1891 0.0945 7.1843 0.7047 1.5130 3.6554 0.0945
0 5.8950 3.1494

0 0 0

Number of errors below 5.0 = 37

Total error = 87.4081

Signature is declared as original

(A forgery is declared as original because of very minute variation in original signature)

CHAPTER 07:

CONCLUSION

7.1. Conclusion:

A novel off-line signature verification algorithm has been presented which uses the soft-computing technique **CLUSTERING**. As clear from the simulation results on Matlab 7.1 environment this algorithm is capable of verifying almost all the signatures. The 42 equi-spaced features are adjusted or updated using the cluster update algorithm and these centres or feature points are trained using the training signatures in the database to avoid interpersonal and intrapersonal errors as much as possible. Despite our best efforts there still are some loop holes in the algorithm, due to which there are some errors in the result.

7.2. Future Work:

- The algorithm discussed in the thesis is not adaptive. That is it has its pre-specified threshold of max error = 155, pre-specified step size in the clustering that is 0.005 etc. All these parameters can be made adaptive, that will adjust them according to the input given to them.
- By using RBFN or ANN this off-line signature verification system can be made even more robust.
- The extraction of exact signature from the signature image sometimes produces error in verifying the signature as seen in the simulation result of person # 04, so a better signature extraction technique can be used.

References: -

1. Neuro-fuzzy and soft computing – by Jang, Sun and Mizutani.
2. Learning Soft Computing – by Vojislav Kecman.
3. Y. Mizukami, M. Yoshimura, H. Miike, and I. Yoshimura, " An off-line signature verification system using an extracted displacement function," Pattern Recognition Letters, 2002, vol. 23
4. R.C., Woods E., 'Digital Image Processing', Addison-Wesley, 1993.
5. R.C., Woods E., 'Digital Image Processing using Matlab', Addison-Wesley, 1993.
6. Qi.Y, Hunt B.R., 'Signature Verification using Global and Grid Features', Pattern Recognition, Vol. 27, No. 12, 1994, pp. 1621-1629.