Invariant Features Comparison in Hidden Markov Model and SIFT for Offline Handwritten Signature Database

Neeraj Shukla
Asst. Professor
Gyan Ganga College of Technology
Tilwara Jabalpur (M.P.) India

Dr. Madhu Shandilya
Associate Professor
Maulana Azad National Institute of Technology , Bhopal (M.P.) India

ABSTRACT
In Handwritten signatures analyzed for forgery have to undergo feature extraction process, due to varied samples in size rotation and intra-domain changes, invariance has to be achieved during feature extraction process; circular Hidden Markov Model with discrete radon transform approach of feature extraction provides invariance. On other hand Scale Invariant Feature Transform (SIFT) has inherent invariant feature extraction approach. This paper compares both approaches on common signature databases for False acceptance rate(FAR),False Rejection Rate(FRR) and Equal Error Rate(EER)

Categories and Subject Descriptors
The Paper deals in **Digital Forensic** category where circular Hidden Markov Model(**HMM**) and Scale invariant Feature Transform(**SIFT**) invariant features are compared for Offline Handwritten Signature verification

General Terms
Here we will be dealing with feature extraction of offline handwritten signature with Discrete Radon Transform (**DRT**) for circular HMM for forgery detection and in second approach scale invariant image features of offline signatures are extracted using **SIFT** and forgery detection is done.

Keywords
Off-line, Signature forgery, Discrete Radon Transform (DRT), Baum-Welch, Viterbi ,Hidden Markov Model (HMM) TP True Positive FP False Positive FN False Negative TN True Negative FAR False Acceptance Rate HMM Hidden Markov Model NN Neural Networks SIFT Scale Invariant Features Transform AER Average Error Rate EER Equal Error Rate FRR False Rejection Rate HSV Handwritten Signature Verification FA False Acceptance SVM Support Vector Machine DoG Difference-of-Gaussian

1. INTRODUCTION

1.1.1 OVERVIEW
The National Check Fraud Center Report of 2000 [1] states that: "cheque fraud and counterfeiting are among the fastest-growing crimes affecting the United States' financial system, producing estimated annual losses exceeding $10 billion with the number continuing to rise at an alarming rate each year."

This system assumes that the signatures have already been extracted from the documents .Methods for extracting signature data from cheque backgrounds can be found in the following papers, [2, 3, 4].

Plamondon and Srihari [5] note that automatic signature verification systems occupy a very specific niche among other automatic identification systems: The features that are extracted from static signature images can be classified as global or local features. Global features describe an entire signature and include the discrete Wavelet transform [7], the Hough transform [8], horizontal and vertical projections [9], and smoothness features [10]. Local features are extracted at stroke and substroke levels and include unballistic motion and tremor information in stroke

segments [11], stroke "elements" [9], local shape descriptors [12], and pressure and slant features [13].

Various pattern recognition techniques have been exploited to authenticate handwritten signatures (see Section 2). These techniques include template matching techniques [7, 9, 11], minimum distance classifiers [10, 12, 14, 15], Neural networks [8, 13, 16], hidden Markov models (HMMs) [17, 18], and structural pattern recognition techniques.

The period from 1989 to 1993 is covered by Leclerc and Plamondon [19] and the period before 1989 by Plamondon and Lorette [20]. Another survey was published by Sabourin et al. in 1992 [21]. A review of online signature verification by Gupta and McCabe in 1998 also includes a summary of some earlier work on the offline case [22].

1.1.2 IMAGE PROCESSING
Each signature is scanned into a binary image at a resolution of 300 dots per inch, after which median filtering is applied for removal of noise. The image dimensions are not normalized because DRT will be used.

The DRT of each signature is calculated. A projection or shadow of the signature at certain angle is represented by each column of the DRT. These projections are processed and normalized which will represent a set of feature vectors, it is also termed as observation sequence of the target signature.

1.2.CALCULATION PROCEDURE OF DRT.

Let us assume that each signature consists of Ψ number of pixels in total and the intensity of ith pixel is denoted by I, i=1,….., Ψ. The DRT is calculated using β non overlapping beams per angle and θ angles in total. The cumulative intensity of the pixels that lie within the jth beam is denoted by Rj. j=1,….., $\beta\theta$. This is

called the jth beam sum. The Discrete Radon Transform can be expressed as follows

$$R_j = \sum_{i=1}^{\Psi} w_{ij} I_i, \quad j = 1,2,\ldots\ldots,\beta\theta, \qquad (1.1)$$

where wij indicates the contribution of ith pixel to the jth beam sum (see figure 1.1).The value of wij is found through two–dimensional interpolation. Each projection therefore contains the beam sums that are calculated at a given angle.

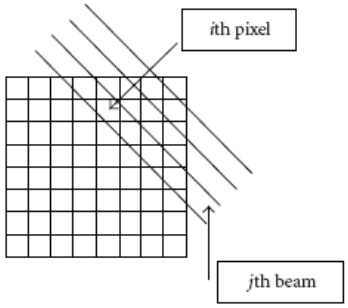

Figure1.1 : Discrete model for the Radon transform with wij≈0.9.

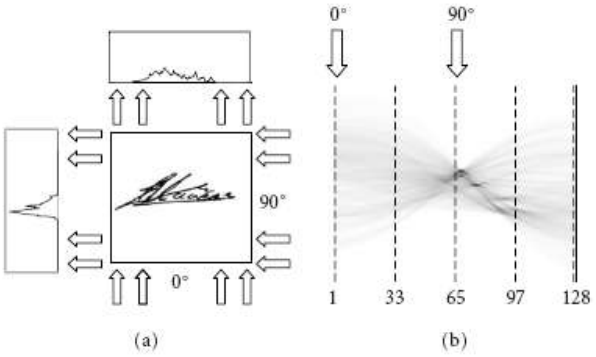

Figure 1.2: (a) A signature and projection calculated at angles of 0o and 90 . (b) The DRT displayed as a gray-scale image. This image has θ=128 columns, where each column represents a projection.

The accuracy of the DRT is determine by θ(the number of angle), β (the number of beams per angle) and the accuracy of the interpolation method.

Note that the continuous form of the Radon transform can be inverted through analytical means . The DRT therefore contains almost the same information as the original image and can be efficiently calculated with an algorithm by Bracewell[23].Our system calculates the DRT at θ angles. These angles are equally distributed between 0o and 180o. A typical signature and its DRT are shown in Figure 2. The dimension of each projection is subsequently altered from β to d.

This is done by first decimating all the zero-valued components from each projection. These decimated vectors are then shrunk or expanded to a length of d through interpolation. Although almost all the information in the original signature image is contained in the projections at angles that range from 0o to 180o, the projections at angles that range from 180o to 360o are also included in the observation sequence. These additional projections are added to the observation sequence in order to ensure that the sequence fits the topology of our HMM (see Section 3.2). Since these projections are simply reflections of the projections already calculated, no additional calculations are necessary. An observation sequence therefore consists of T = 2θ feature vectors, that is, XT1 = {x1, x2, . . . , xT}. Each vector is subsequently normalized by the variance of the intensity of the entire set of T feature vectors. Each signature pattern is therefore represented by an observation sequence that consists of T observations, where each observation is a feature vector of dimension d. The experimental results and computational requirements for various values of d and θ are discussed in Sections 6 . The DRT, as a feature extraction technique, has several advantages. Although the DRT is not a shift invariant representation of a signature image, shift and scale invariance is ensured by the subsequent image processing. Each signature is a static image and contains no dynamic information. Since the feature vectors are obtained by calculating projections at different angles, simulated time evolution is created from one feature vector to the next, where the angle is the dynamic variable. This enables us to construct an HMM for each signature (see Section 3). The DRT is calculated at angles that range from 0o to 360oand each observation sequence is then modeled by an HMM of which the states are organized in a ring (see Section 3.2). This ensures that each set of feature vectors is rotation invariant. Our system is also robust with respect to moderate levels of noise. These advantages are now discussed in more detail.

1.3 Noise

We explained earlier in this section that the zero-valued components of each projection are decimated before the remaining non-zero components are shrunk or expanded through interpolation. In this way, a feature vector with the required dimension is obtained. The decimation of the zero-valued components ensures that moderate levels of noise (which are represented by a few additional small-valued components within certain projections) are "attached" to the other nonzero components before the decimated vector is shrunk or expanded. Since the dimension of the feature vectors are high compared to the number of these additional components, the incorporation of these components has little effect on the overall performance of the system.

1.4 Shift invariance

Although the DRT is not a shift invariant representation of a signature image, shift invariance is ensured by the subsequent image processing. The zero-valued components of each projection are decimated and the corresponding feature vector is constructed from the remaining components only

1.5 Rotation invariance

The DRT is calculated at angles that range from 0o to 360o and each set of feature vectors is then modeled by an HMM of which the states are organized in a ring (see Section 3.2). Each

signature is therefore represented by a set of feature vectors that is rotation invariant

1.6 Scale invariance

For each projection, scale invariance has to be achieved in the direction perpendicular to the direction in which the image is scanned, that is, perpendicular to the beams, and in the direction parallel to the beams. Scale invariance perpendicular to the beams is ensured by shrinking or expanding each decimated projection to the required dimension. Scale invariance parallel to the beams is achieved by normalizing the intensity of each feature vector. This is achieved by dividing each feature vector by the variance of the intensity of the entire set of feature vectors.

2.SIGNATUREMODELLING

We use a first-order continuous observation HMM to model each writer's signature. For a tutorial on HMMs, the reader is referred to a paper by Rabiner [24] and the book by Deller et al. [25].

2.1. Notation

We use the following notation for an HMM λ.

(1) We denote the N individual states as S = {1, s2, . . . , sN} (2) and the state at time t as qt .

(2) The initial state distribution is denoted by π = {πi}, where πi = P_q1 = si_, i = 1, . . . , N. (1.3)

(3) The state transition probability distribution is denoted by A = {ai, j}, where ai, j = P_qt+1 = sj | qt = si_, i = 1, . . . ,N, j = 1, . . . , N. (1.4)

(4) The probability density function (pdf), which quantifies the similarity between a feature vector x and the state sj, is denoted by f _x|sj , λ_, j = 1, . . . , N. (1.5)

2.2. HMM topology

We use an HMM, the states of which are organized in a ring (see Figure 2.1).

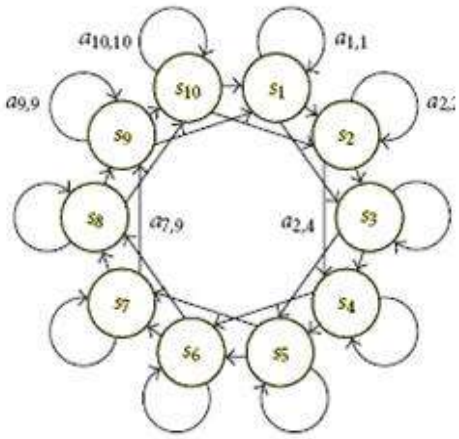

Figure 2.1: An example of an HMM with a ring topology. This model has ten states with one state skip.

Our model is equivalent to a left-to-right model, but a transition from the last state to the first state is allowed. Since the HMM is constructed in such a way that it is equally likely to enter the model at any state, and the feature vectors are obtained from all the projections, that is, the projections calculated at angles ranging from 0o to 360o the ring topology of our HMM guarantees that the signatures

are rotation invariant. Each state in the HMM represents one or more feature vectors that occupy similar positions in a d-dimensional feature space. This implies that the HMM

groups certain projections (columns of the DRT) together. It is important to note that this segmentation process only takes place after some further image processing has been conducted on the original projections.

2.3. Training Using Viterbi Algorithm:

Each model is trained using the Viterbi re-estimation technique. The dissimilarity between an observation sequence X and a model λ can therefore be calculated as follows (see [4]):

d(X, λ) = −ln _ f (X|λ)_. (6) In real-world scenarios, each writer can only submit a small number of training samples when he or she is enrolled into the system. Since our algorithm uses feature vectors with a high dimension, the re-estimated co-variance matrix of the pdf for each state is not reliable and may even be singular. A Mahalanobis distance measure can therefore not be found. Consequently, these covariance matrices are not re-estimated and are initially set to 0.5I, where I is the identity matrix. Only the mean vectors are re-estimated, which implies that the dissimilarity values are based on an Euclidean distance measure. We assume that training signatures, genuine test signatures, and forgeries are available for only a limited number of writers, that is, for those writers in our database. No forgeries are used in the training process since our system aims to detect only skilled and casual forgeries, and these type of forgeries are not available when our system is implemented. The genuine test signatures and forgeries are used to determine the error rates for our system (see Section 3). Assuming that there are W writers in our database, the training signatures for each writer are used to construct an HMM, resulting in W models, that is {λ1, λ2, . . . , λW}.

When the training set for writer w is denoted by {X(w) 1 ,X(w) 2 , . . . ,X(w) Nw }, where Nw is the number of samples in the training set, the dissimilarity between every training sample and the model is used to determine the following statistics for the writer's signature:

$$\mu_w = \frac{1}{N_w}\sum_{i=1}^{N_w} d\left(X_i^{(w)}, \lambda_w\right), \qquad (2.1)$$

$$\sigma_w^2 = \frac{1}{N_w - 1}\sum_{i=1}^{N_w}\left(d\left(X_i^{(w)}, \lambda_w\right) - \mu_w\right)^2. \qquad (2.2)$$

2.4. VERIFICATION

When a system aims to detect only random forgeries, subsets of other writers' training sets can be used to model "typical" forgeries. This is called "impostor validation" and can be achieved through strategies like test normalization (see [26]). These techniques enable one to construct verifiers that detect random forgeries very accurately (see [7, 8]). Since we aim to detect only skilled and casual forgeries, and since models for these forgeries are generally unobtainable, we are not able to utilise any of these impostor validation techniques. We also do not use any subset of genuine signatures for validation purposes. Our verifier is constructed as follows. When a claim is made that the test pattern X(w) Test belongs to writer w, the pattern is first matched with the model λw through Viterbi alignment. This match is quantified by f (X(w) Test|λw). The dissimilarity between the test pattern and the model is then calculated as follows (see [4]):

$$d\left(X_{\text{Test}}^{(w)}, \lambda_w\right) = -\ln\left(f\left(X_{\text{Test}}^{(w)} | \lambda_w\right)\right). \qquad (2.3)$$

2.5. Experimental setup

We consider 30 genuine signatures, 6 skilled forgeries, and 6 casual forgeries for each writer. For each writer, 10 genuine signatures are used for training and 20 for testing. No genuine signatures are used for validation purposes.

2.6 Results

Let ℓ denote the number of allotted forward links in our HMM. Figure 4 shows the FRR and FAR as functions of our threshold parameter τ Є [−0.1, 1], when d = 512, θ = 128, N = 64, and ℓ = 1. The FRR, the FAR for a test set that

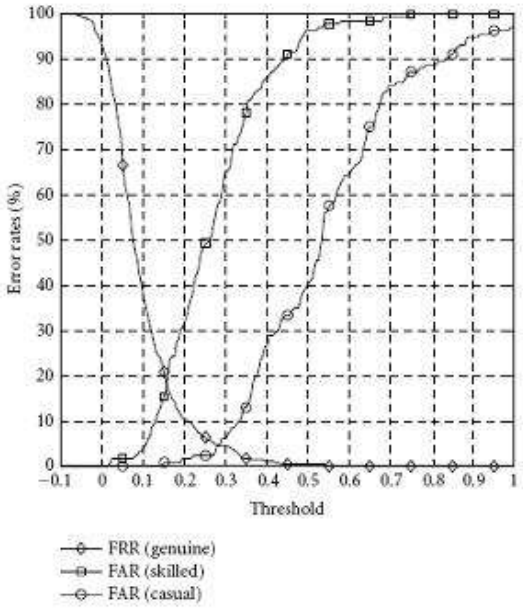

FIGURE 2.2 The Stellenbosch data set. Graphs for the FRR and the FAR when $d = 512$, $\Theta = 128$, $N = 64$, and $\ell = 1$.

Figure 2.2 contains only skilled forgeries, and the FAR for a test set that contains only casual forgeries are plotted on the same system of axes. When, for example, a threshold of τ = 0.16 is selected, equation (11) implies that all the test patterns for which d(X(w) Test, λw) ≥ 1.16μw are rejected—the other patterns are accepted. When only skilled forgeries are considered, this threshold selection will ensure an EER of approximately 18%. When only casual forgeries are considered, our algorithm achieves an EER of 4.5%. Table 1 tabulates the EER as well as a local FRR and FAR, for various values of d, θ, N, and ℓ. It is clear that when the dimension of the feature vectors is decreased from d = 512 to d = 256 or even to d = 128, the performance of the system is not significantly compromised. The performance of our system is generally enhanced when the number of feature vectors, that is, T = 2θ, or the number of states in the HMM, that is, N, is increased. The best results are obtained when only one forward link is allowed in the HMM, that is, when ℓ = 1.

TABLE 2: The Stellenbosch data set: summary of results.

d	Θ	N	ℓ	FRR (%)	FAR (%)	EER (%)
512	128	64	1			
Skilled forgeries				10.2	25.4	**17.9**
Casual forgeries				0.2	32.6	**4.5**
512	128	64	2			
Skilled forgeries				10.2	25.7	18.5
Casual forgeries				0.2	34.1	4.6
512	128	64	4			
Skilled forgeries				10.2	25.9	18.2
Casual forgeries				0.2	35.6	4.6
512	128	32	1			
Skilled forgeries				10.2	25.5	19.2
Casual forgeries				0.2	34.8	5.4
512	128	16	1			
Skilled forgeries				10.2	32.1	20.7
Casual forgeries				0.2	43.1	6.2
256	128	64	1			
Skilled forgeries				10.2	25.3	**17.7**
Casual forgeries				0.2	32.6	**4.5**
256	128	32	1			
Skilled forgeries				10.2	26.4	19.4
Casual forgeries				0.2	35.6	5.4
256	64	32	1			
Skilled forgeries				10.2	25.4	18.4
Casual forgeries				0.2	34.8	5.3
128	64	32	1			
Skilled forgeries				10.2	24.6	**18.3**
Casual forgeries				0.2	34.8	**5.4**
128	32	16	1			
Skilled forgeries				0.2	38.7	22.0
Casual forgeries				0.2	48.4	6.2

Table 1.1 Summary of Results

3 SIFT Related Work

Proposed by David Lowe, Scale Invariant Features Transform (SIFT) is used to extract distinctive invariant features from images [28]. The SIFT algorithm is robust for identifying stable key locations in the scale- space of a grey scale image [28][52]. It uses the following four steps to extract the set of descriptors from a given image [28].

(i) Scale-Space extrema detection.

(ii) Accurate Keypoint localisation.

(iii) Orientation assignment.

(iv) Keypoint description.

Step 1: Scale-Space extrema detection involves searching over all scales and location of the sig- nature image to detect key points of all sizes. This is done using a difference-of-Gaussian (DoG) function to identify potential interest points that are invariant to scale and orientation [52]. For each octave of scale space, the image is convolved with Gaussian functions producing a set of scale space images. Adjacent Gaussian images are subtracted to produce difference-of –Gaussian images. After each octave the Gaussian image is halved and the process is repeated. Figure 2.1 illustrates the blurred images at different scales and the computation of difference -of- Gaussian (DoG). The Scale-space of a signature image is defined as the function L(x,y,α),which is convolution of a variable scale Gaussian G(x,y,α) with an input signature image I(x,y) as follows [28]:

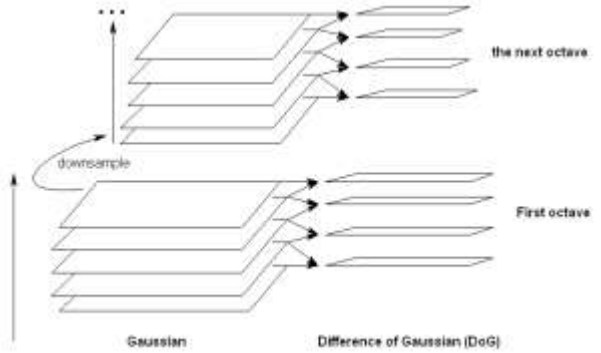

Figure 3.1: Difference -of- Gaussian computation.

$L(x,y,\alpha) = G(x,y,\alpha) * I(x,y)$ (3.1)

where * is the convolution in the x and y directions, and

$G(x,y,\alpha)= 1\ (2\pi a2)1/2\ \exp(-x2+y2\ 2a2\ )$ (3.2)

The difference between two nearby scales, $D(x,y,\alpha)$, separated by a constant multiplicative factor k is given by

$D(x,y,\alpha) = (G(x,y,k\alpha) - G(x,y,\alpha)) * I(x, y)$ (3.3)

$= L(x,y,k\alpha) - L(x,y,\alpha)$ (3.4)

The keypoints are identified as local maxima and minima of the DoG signature images across scale. Each pixel in the DoG is compared to other 8 neighbouring pixels at the same scale and 9 corresponding neighbours at the neighbouring scales. If the keypoint is the local maxima or minima, it is selected as a candidate keypoint. Figure 2.2 illustrates detecting the maxima and minima of difference-of-Gaussian in scale space.

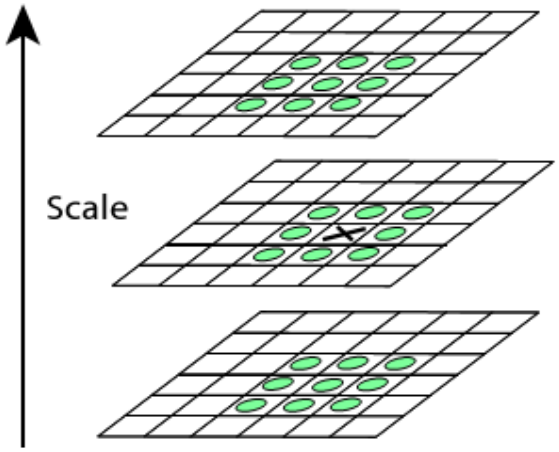

Figure 3.2: Scale space extrema detection (Reproduced from [28]).

Step 2: Accurate keypoint localisation. For each candidate keypoint identified, the interpolation of nearby data is used to accurately determine its point. Keypoints with low contrast (sensitive to noise) are dropped together with the responses poorly localised along the edges.

Step 3: Orientation Assignment. Each keypoint is assigned one or more orientations based on local image gradients directions. To determine the keypoint orientation, a gradient orientation his- togram is computed in the neighborhood of the keypoint using the Gaussian image at the closest scale to the keypoints. The contribution of each neighboring pixel is weighted by the gradient magnitude and a Gaus- sian window with α set to be 1.5 times the scale of the keypoint. This contributes to stability [28]. Peaks at the histogram are correspondent with dominant orientation. Any keypoint that is within 80% of the highest peak is used to create a separate keypoint. The orientation assignment of each keypoint is obtained by computing the gradient magnitude M(x,y) and orientation θ(x, y)of the scale space for the scale of that keypoint:

$$M(x,y)=\sqrt{(K(x+1,y)-K(x-1,y))2-(K(x,y+1)-K(x,y-1))2} \quad (3.5)$$

and

$$\theta(x,y)=\arctan \frac{K(x,y+1)-K(x,y-1)}{K(x+1,y)-K(x-1,y)} \quad (3.6)$$

All the properties of the keypoint are measured relative to the keypoint orientation. This caters for rotation invariance.

Step 4: Keypoint Description. Local image gradients are measured at the selected scale in the region around each key point and transformed into a representation that allows local shape distor- tion and change in illumination. When the keypoint orientation is selected, feature descriptors are computed as a set of orienta- tion histograms on 4*4 pixel neighborhoods. The orientation histograms are relative to the key- point orientation, and the orientation data comes from the Gaussian image closest in scale to the keypoints scale. The contribution of each pixel is weighted by the gradient magnitude and by a Gaussian with α 1.5 times the scale of the keypoint. Histograms contain 8 bins each and each descriptor contains an array of 4 histograms around the keypoint. This gives a SIFT feature with 4*4*8=128 values. This vector is normalized to enhance invariance to illumination.

SIFT features have the following advantages compared to other shape descriptors [28].

(i) Locality-Features detected are local and robust to clutter and occulsion.

(ii) Distinctiveness-Individual features can be matched to a large database.

(iii) Quantity -Many features can be generated even for small objects.

(iv) Efficiency for real time performance.

(v) Extensibility -They can be extended to different dimensions each with added robustness.

SIFT features have been used in pattern recognition and classification, mostly in object recognition. The work of Kim et al [53] uses SIFT features for robust digital watermarking. In [54] , the SIFT algorithm is used for face authentication using frontal view templates and evaluated for recognition of graffiti tags in [55] both with good results. Dlagnekov in his thesis used SIFT features for car make and model recognition with 89.5% true recognition rate [56]. More recently, use of SIFT features in fingerprint verification has been investigated [57]. Unlike these SIFT related work where the verification models have landmark features that have no intra class variability e.g. the location of the mouth and eyes in frontal view face authentication and minutiae points in fingerprint verification, which makes it easier to compute the nearest neighbours from these invariant points and do one to one mapping between the training class and the test class. Signatures have natural variance even among genuine signatures.

3.1 Introduction of Methodology

Computer vision is often concerned with recognition of objects in a manner invariant to scale, pose, illumination and affine distortion. The SIFT algorithm takes an image and transforms it into a collection of local features where each of these feature vectors are distinctive and invariant to any scaling, rotation or translation of the image. In this project the SIFT features were considered. The implementation was done in MATLAB 6.0. The approach taken is a two step process with signature enrolment and verification. The forged signatures in the test set were generated by imitating the genuine signatures for each class on a piece of paper. The forgery was done by two people each generating a sample of three forged signatures per class which were given to a third party to chose one forgery which closely resembles the genuine set. Each forged signature was also scanned, cropped and stored in portable network graphic format. The results obtained from SIFT based verifier was compared with the results from human experts. Our original aim to use benchmark datasets from other research studies was not possible due to lack of cooperation and unavailability of online public datasets which are purely for offline handwritten signatures.

3.2 Steps Used in Offline Handwritten Signature Verification

The approach used for offline handwritten signature verification was broadly divided into two steps, signature enrolment and signature verification. Signature enrolment had four sub steps namely image pre-processing, extraction of SIFT features from signatures, calculation of Eu- clidean distances between images and creation of the known class signatures template. Signa- ture verification had two sub steps namely outlier detection and comparison of test signature with known set so as to make a decision whether it is a genuine signature or not.

3.3 Signature Enrolment

Signature enrolment involved preparation of signatures, extraction of SIFT features and registra- tion of signatures images and their SIFT features in the system.

3.3.1 Image Pre-Processing

The images used were signatures and were extracted from documents through scanning and crop- ping. A random sample of 18 signers was used, each signer contributed a sample of 3 signaturesgiving a total of 54 genuine signatures for the training

set. The test set consisted of 18 genuine signatures and 18 forged signatures giving a total of 36 signatures for the test set. A database of 90 signatures was used in overall i.e. the training set and test set. Signature images were stored in portable network graphic (PNG) format. These images were converted to greyscale for further processing.

3.3.2 Extraction of SIFT Features From Signatures

This involved identifying stable shape descriptors from the pre processed signature image as de- scribed in Section 2.8 . The implementation that was used for extracting SIFT features was adopted from a MATLAB function written by El-Maraghi [58]. Figure 3.1 shows an example of scale space Gaussian images for one of the signatures in the test set. Figure 3.2 shows a sample signature and its keypoints and their orientation.

Figure 3.3: Example of space scale Gaussian images.

Figure 3.4: Example of a signature with extracted SIFT features.

3.3.3 Calculation of Euclidean Distances

This involved calculation of the Euclidean distances between the SIFT features of two given sig- nature images to measure the variability between them. The motivation to use Euclidean distance as a measure of variability between images is derived from its success in object recognition [52] and lately in fingerprint verification [57]. Say we have two signatures A and B. Let Ai be the ith keypoint in signature A and Bj be the jth keypoint in signature B. The distance D(Ai,Bj) was calculated as the Euclidean distance between Ai and Bj. Ka, Kb are the number of keypoints in signature A and B respectively. The distance measure D(Ai,B) was taken as the average Euclidean distance from the ith keypoint in signature A to all the keypoints of signature B. The image distance between signature A and signature B is given by : D(A, B) = 1 Ka Ka∑i=1D(Ai,B)(3.7)

3.3.4 Creation of the Known Signature Template.

The implementation focused on upholding anonymity of the signers. Only the signatures and ar-bitrary writer IDs were used. For each known writer, a sample of three signatures say A, B and C were taken to cater for intra-personal variations. A template was generated as a MATLAB file and stored. The template has the following:

(i) Writer ID.

(ii) The Euclidean distances between keypoints i.e. D(Ai,B), D(Ai,C), and D(Bj,C).

(iii) The distances between the Signature images i.e. D(A,B), D(A,C) and D(B,C).

(iv) Intra-class thresholds: The maximum among D(A, B),D(A, C) and D(B,C) i.e.

max(D(A, B),D(A, C),D(B,C)). The minimum among D(A, B),D(A, C) and D(B,C)

i.e. min(D(A, B),D(A, C),D(B,C)). The average on D(A, B),D(A, C) and D(B,C)

i.e. avg (D(A, B),D(A, C),D(B,C)). The range on maximum intra-class distance given

by max(D(A, B),D(A, C),D(B,C)) ą 0.05. The range on minimum intra -class distance

given by min(D(A, B),D(A, C),D(B,C)) ą 0.05.

Figure 3.5 is an example of a sample of three genuine signatures of a known writer taken to

cater for intra-personal variation.

Figure 3.5: Example of intra-personal variation.

Figure 3.6 Summarizes the signature enrolment stage.

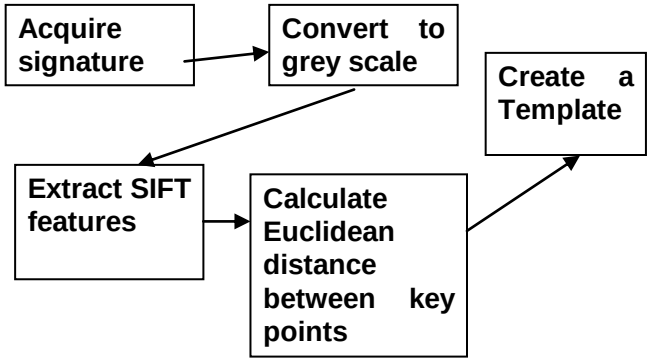

Figure 3.6: Steps in signature enrolment.

3.4 Signature Verification

Verification is the process of testing whether a claimed signature is of the same (class) writer as the set of signatures enrolled in

the system for that class. Verification involved loading the template MATLAB file enrolled in the system and comparing its stored parameters with those calculated by the outlier detection process.

3.4.1 Outlier Detection

Given a test signature say T claimed to be of a particular writer, the Euclidean distances were

calculated between the test signature and each of the three sample signatures (as discussed in Sub- section 3.3.3) resulting to distances between the images i.e. D(T,A), D(T,B) and D(T,C).The inter-class thresholds, max(D(T,A),D(T,B),D(T,C)), min(D(T,A),D(T,B),D(T,C)),

avg (D(T,A),D(T,B),D(T,C)) are computed.

3.4.2 Comparison and Decision Criteria

The comparison between the distance parameters of the SIFT features of the claimed test sig-nature was done with those of the stored template. Each decision criteria was a binary clas-sification and was taken independently. We let W be (D(T,A),D(T,B),D(T,C)) and Z be(D(A, B),D(A, C),D(B,C)).

Test 1: Comparing inter-class maximum distance with intra-class maximum distance asthreshold.

We classify T as genuine if the condition

max(Z) > max(W)(3.8)

holds, otherwise we classify T as not genuine.

Test 2: Comparing average of inter-class distances with the average of intra-class distance asthreshold.

We classify T as genuine if the condition

avg (Z) > avg (W)(3.9)

holds, otherwise we classify T as not genuine.

Test 3: Comparing inter-class minimum distance with intra-class minimum distance as thresh-old.

We classify T as genuine if the condition

min(Z) > min(W)(3.10)

holds, otherwise we classify T as not genuine.

Test 4: Using a range of 0.05 on the maximum intra-class distance as a threshold and comparing with inter-class maximum distance.

We classify T as genuine if the conditionmax(Z) ą 0.05 > max(W)(3.11)

holds, otherwise we classify T as not genuine.

Test 5: Using a range of 0.05 on the minimum intra-class distance as a threshold and comparing with inter-class minimum distance.

We classify T as genuine if the conditionmin(Z)) ą 0.05 > min(W))(3.12)

holds, otherwise we classify T as not genuine.

Test 6: Using a range of 0.05 on both the minimum intra-class distance and minimum intra-class distance as a threshold such that the minimum and maximum inter- class distanceshould lie within that range.

We classify T as genuine if the condition max(Z) ą 0.05 > max(W) and min(Z) ą 0.05 > min(W)(3.13)

holds, otherwise we classify T as not genuine.

Figure 3.7 summarizes the signature enrolment and verification .

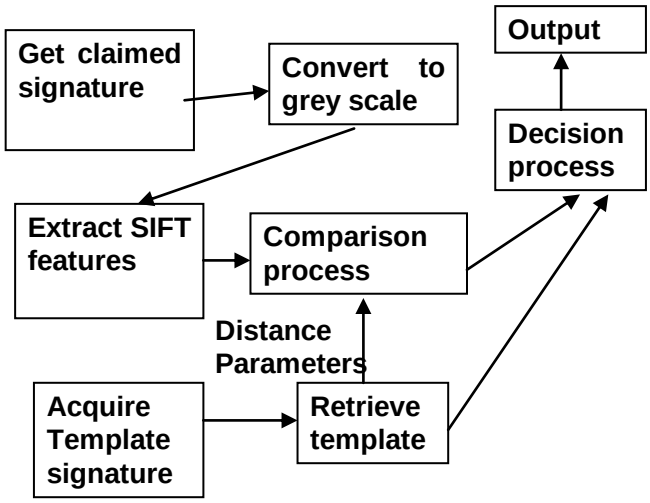

Figure 3.7: Flowchart showing signature enrolment and verification.

3.5 Measurement of the Signature Verifier Accuracy

To measure the accuracy of the verifier, a set consisting of genuine signatures and forged signatures was used and various performance statistics were used. These statistics are standard in machine learning literature, see example in Section 5.7 of [59].

(i) **True Positive** (TP) - A classification is a true positive if the signature is genuine (of known writer) and the output of the verifier ascertains that.

(ii) **False Positive** (FP) - A classification is a false positive if the signature is forged and the

output of the verifier claims that it is genuine.

(iii) **True Negative** (TN) - A classification is a true negative if the signature is forged and the

output of the verifier ascertains that.

(iv) **False Negative** (FN) - A classification is a false negative if the signature is genuine (of

known writer) and the output of the verifier claims that it is forged.

(v) **The** sensitivity is the proportion of actual positives (genuine signatures) which are correctly identified as positives. which is given by:

Sensitivity =TP TP + FN (3.8)

(vi) **The specificity** is the proportion of negatives (forgeries) which are correctly identified,

which is given by:Specificity =TN TN + FP (3.9)

The test for accuracy of the system is summarised in Figure 3.6 :

		Actual Condition(Truth)	
		(+ve)Genuine	(-ve)Forgery
OUTPUT OF THE SYSTEM	(+ve)Genuine	TP	FP
	(-ve)Forgery	FN	TN

Figure 3.8: Confusion matrix for analysing accuracy.

4 RESULTS

4.1 Introduction

To measure the accuracy of the SIFT based verifier, a set consisting of genuine signatures and forged signatures was used. In total 90 signatures were used. The training set had 54 genuine signatures for creating the known signature templates. A test set consisted of a total of 36 signatures (18 genuine signatures and 18 forged signatures). For each class of known signatures containing three sample signatures, a genuine and a forged signature were tested independently. The overall performance of the SIFT based classifier was measured in terms of the number of genuine and forged signatures it can correctly classify in the test set.

4.2 Examples of Verified Signatures

In this Section we present examples of verified signatures. Figure 4.1 shows signatures 16.png, 17.png and 18.png from the same known writer(same class) and were used as the training set for this class to create a template. The signatures 19.png in Figure 4.2 was the test signature. Using all the five tests described in Subsection 3.4.2, signature 19.png was correctly identified as gen- uine. Table 4.1 shows the image distances between the set of known signatures 16.png, 17.png and 18.png. The intra class maximum, max(D(16,17),D(17,18),D(16,18)) = 1.1710 is greaterthan the inter class maximum max(D(16,19),D(17,19),D(18,19)) = 1.0700. The intra class average, avg (D(16,17),D(17,18),D(16,18)) = 1.1293 is greater than the inter class average avg (D(16,19),D(17,19),D(18,19)) = 1.0497, the intra class range on maximumm intra class distances is 1.2210 is also greater than inter class maximum max(D(16,19),D(17,19),D(18,19))

= 1.0700. The intra-class minimum min(D(16,17),D(17,18),D(16,18)) = 1.1069 is greater than inter class minimum distance which is 1.0382. Also the range on minimum, min(D(16,17),D(17,18),D(16,18))- 0.05= 1.0569 is also greater than inter class minimum. Hence based on all the tests signature 19.png is correctly classified as genuine. Table 4.2 shows the inter- class distances between the test signature 19.png and the template of knowns.

Figure 4.1: Example 1 of genuine signatures of a known writer.

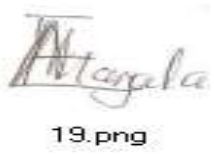

Figure 4.2: Test signature correctly classfied as genuine by all the tests.

Table 4.1: Image distances set of known signatures 16.png, 17.png and 18.png.

Signatures	Distance description	Image distance
16.png,18.png	D(16,18)	1.1069
17.png,18.png	D(17,18)	1.1710
16.png,17.png	D(16,17)	1.1099

Table 4.2: Image distances between test signature 19.png and set of known signatures.

Signatures	Distance description	Image distance
16.png,19.png	D(16,19)	1.0411
17.png,19.png)	D(17,19	1.0700
18.png,19.png	D(18,19)	1.0382

Figure 4.3 shows signatures 41.png, 42.png and 43.png from the same known writer and were

used as the training set for this class to create a template. Using this template, signature 45.png shown in Figure 4.4 was correctly classified as a forgery by all the tests. Table 4.3 shows the intra - class distances between signatures 41.png, 42.png and 43.png. Table 4.4 shows the inter – class distances between known signatures 41.png, 42.png, 43.png and test signature 45.png.

Figure 4.3: Example 2 of genuine signatures of a known writer.

Figure 4.4: Test signature correctly classfied as forgery by all the tests.

Table 4.3: Image distances set of known signatures 41.png, 42.png and 43.png.

Signatures	Distance description	Image distance
41.png,42.png	D(41,42)	1.0538
41.png,43.png	D(41,43)	1.0538
42.png,43.png	D(42,43)	1.1028

Table 4.4: Image distances between test signature 45.png and set of knowns 41.png, 42.png and 43.png.

Signatures	Distance description	Image distance
41.png,45.png	D(41,45)	1.2012
42.png,45.png	D(42,45)	1.3967
43.png,45.png	D(43,45)	1.0539

4.3 Results from the Proposed Method

MATLAB scripts were used to detect false positives, true positives, true negatives, true positives and to calculate the sensitivity and the specificity. Sensitivity is proportion of genuine signatures the classifier is able to correctly identify as genuine from the test set and the specificity is the proportion of the forgeries the classifier is able to correctly classify as forgeries from the test set.The following statistics were obtained.

4.3.1 Maximum Distance

The specificity of 38.89% was obtained; which is the proportion of forgeries the classifier was able to identify from the testing set and the sensitivity of 77.78% was also obtained; which is the pro- portion of genuine signatures the classifier was able to correctly identify after using the condition set in Equation 3.2, that is comparing the maximum intra-class distance with maximum inter-class distance. This means the comparison between the maximum intra - class distance and maximum inter - class distance was better in identifying genuine signatures than in detecting forgeries. Table 4.5 shows the performance statistics obtained by the classifier using maximum class distances. Table 4.5: Performance statistics obtained by the classifier using maximum class distances.

TP	**14**	**FP**	**11**
TN	**7**	**FN**	**4**

4.3.2 Average Distance

Using the condition set in Equation 3.3, that is comparing the average intra-class distance with average inter-class distance. The specificity of 50% was obtained, which is the proportion of forged signatures correctly identified from the test set and the sensitivity of 44.444% was also obtained, that is the proportion of genuine signatures correctly identified. From these performance statistics it shows the average test was poor and random in both detecting the forged signatures and identifying the genuine signatures. Table 4.6 shows the performance statistics obtained by the classifier using average class distances. Table 4.6: Performance statistics obtained by the classifier using average class distances.

TP	**8**	**FP**	**9**
TN	**9**	**FN**	**10**

4.3.3 Minimum Distance

The specificity of 38.889% and the sensitivity of 44.444% were obtained after using the condition set in Equation 3.4, that is comparing the minimum intra-class distance with minimum inter-class distance. Similar to the average test, the minimum distance test performed poorly in both detecting the forged signatures and identifying the genuine signatures. Table 4.7 shows the performance statistics obtained by the classifier using minimum class distances. Table 4.7: Performance statistics obtained by the classifier using minimum class distances.

TP	**7**	**FP**	**10**
TN	**8**	**FN**	**11**

4.3.4 Range of q0.05 on Maximum Distance

The specificity of 33.3% and the sensitivity of 88.8% were obtained after using the condition set

in Equation 3.5, that is a range of 0.05 on the maximum intra-class distance and setting it as a threshold and comparing it with the maximum inter-class distance. This test was the best in terms of sensitivity i.e. was able to correctly classify highest number of genuine signatures from the test set and the poorest in terms of specificity i.e. identifying forged signatures. Table 4.8 shows the performance statistics obtained by the classifier using the range test on maximum intra class distance.Table 4.8: Performance statistics obtained by the classifier using the range test on maximum class distances.

TP	**16**	**FP**	**15**
TN	**3**	**FN**	**2**

4.3.5 Range of q0.05 on Minimum Distance

The specificity of 72.2% and the sensitivity of 50% were obtained after using the condition set in Equation 3.6, that is a range of 0.05 on the minimum intra-class distance and setting it as a threshold and comparing it with the minimum inter-class distance. This test was the best in identifying the forged signatures from the test set. Table 4.9 shows the performance statistics obtained by the classifier using the range test on minimum intra class distance .Table 4.9: Performance statistics

obtained by the classifier using the range test on minimum class distances.

TP	**9**	**FP**	**5**
TN	**13**	**FN**	**9**

4.3.6 Range of q0.05 on Maximum Distance and Range of q0.05 on Mini- mum Distance

The specificity of 55.5% and the sensitivity of 77.78% were obtained after using the condition

set in Equation 3.7, that is a a range of 0.05 on both the minimum and maximum intra-class distances and setting them as a threshold. Table 4.10 shows the performance statistics obtained by the classifier using the range on both minimum and maximum intra-class distances. A good classifier should have high rates of both specificity and sensitivity. It should be able to correctly classify high proportion of genuine signatures from the test set and also detect high proportion of forged signatures as forgeries in the same test set. From the performance statistics, this test compared to the rest had high rates on both specificity and sensitivity and was considered for comparison with human experts. Table 4.10: Performance statistics obtained by the classifier using the range test on both minimum and maximum class distances.

TP	**14**	**FP**	**8**
TN	**10**	**FN**	**4**

5. CONCLUSIONS AND AREAS OF FURTHER RESEARCH

5.1 Conclusions

The objective of this project was mainly to offer an efficient and economically viable offline hand- written signature verifier. In order to meet the objective various existing methods of offline hand- written signature verification were reviewed and SIFT features were decided as robust image de- scriptors. A database of signatures was collected consisting of known writers' signatures and forgeries. The efficiency of the verifier was tested and specificity and the sensitivity were measured for each test taken. It was noted that some writers have large discrepancies between three of their sample signatures such that even a forgery may fall within the intra class distances which may result to a false negative notification this might have been caused by physiological factors. A good classifier should have high rates of specificity and sensitivity. To be able to have an efficient classifier we picked the test that had high rates of both specificity and sensitivity. The optimal condition was given by Equation 3.7 that is, using a range of 0.05 on both the minimum intra-class distance and minimum intra-class distance as a threshold such that the minimum and maximum inter- class distance should lie within that range. Though originally designed for object recognition, the use of SIFT features for signature verification had not been systematically investigated before. The performance statistics obtained from this test showed that SIFT features can be used with Euclidean distances for offline handwritten verification. Although this research is a good start to SIFT based handwritten signature verification it can be extended to evaluate other image similarity measures.

5.2 Areas of Further Research

The problem of handwritten signature verification was addressed from an offline point of view in the experiments. Many areas of study related to SIFT features and various distance measures are still open.

5.2.1 Alternative Distance Measures

Use of SIFT features as signature descriptors and other distance measures could be interesting. Chernoff-Bhattacharya distance, has been successfully used to measure discrminability in hand-written numeral recognition [60] could be evaluated in HSV problems. Mahalanobis distance is another measure that can be used to find patterns in SIFT features . Unlike the Euclidean distance that uses the mean vector, Mahalanobis distance uses both the mean vector and the full covariance matrix which can an efficient measure of variability among signatures. If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. Detailed explanations of the Chernoff-Bhattacharya distance and Mahalanobis distance can be found in Chapter 6 of [61]. The experiments can also be extended to combine two or more of these distance measures and compare their efficiency.

5.2.2 SIFT Features and Online Handwritten Signature Verification

Since online handwritten signature verification problems involves descriptors like velocity, acceleration and capture time of each point on the signature trajectory. Future work could evaluate inclusion of SIFT features as image descriptors and various distance measures discussed above in online handwritten signature verification problems.

ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

REFERENCES

[1] National Check Fraud Center, National Check Fraud Center Report, 2000.

[2] S. Djeziri, F. Nouboud, and R. Plamondon, "Extraction of signatures from cheque background based on a filiformity criterion," IEEE Trans. Image Processing, vol. 7, no. 10, pp. 1425– 1438, 1998.

[3] A. L. Koerich and L. L. Lee, "Automatic extraction of filledin information from bankchecks based on prior knowledge about layout structure," in Advances in Document Image Analysis: First Brazilian Symposium, vol. 1339 of Lecture Notes in Computer Science, pp. 322–333, Curitiba, Brazil, November 1997

[4] J. E. B. Santos, F. Bortolozzi, and R. Sabourin, "A simple methodology to bankcheck segmentation," in Advances in Document Image Analysis: First Brazilian Symposium, vol. 1339 of Lecture Notes in Computer Science, pp. 334–343, Curitiba, Brazil, November 1997

[5] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: a comprehensive survey," IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63–84, 2000.

[6] J. G. A. Dolfing, Handwriting recognition and verification. A hidden Markov approach, Ph.D. thesis, Eindhoven University of Technology, 1998.

[7] P. S.Deng, H. Y. M. Liao, C.-W.Ho, and H.-R. Tyan, "Waveletbased off-line handwritten signature verification," Computer Vision and Image Understanding, vol. 76, no. 3, pp. 173–190, 1999.

[8] T. Kaewkongka, K. Chamnongthai, and B. Thipakorn, "Offline signature recognition using parameterized hough transform," in Proc. 5th International Symposium on Signal Processing and Its Applications, pp. 451–454, Brisbane, Australia, 1999.

[9] B. Fang,C.H. Leung, Y. Y. Tang, K. W. Tse, P. C. K. Kwok, and Y. K. Wong, "Off-line signature verification by the tracking of feature and stroke positions," Pattern Recognition, vol. 36, pp. 91–101, 2003.

[10] B. Fang, Y. Y. Wang, C. H. Leung, et al., "Offline signature verification by the analysis of cursive strokes," International Journal of Pattern Recognition and Artificial Intelligence, vol. 15, no. 4, pp. 659–673, 2001.

[11] J. K. Guo, D. Doermann, and A. Rosenfeld, "Forgery detection by local correspondence," International Journal of Pattern Recognition and Artificial Intelligence, vol. 15, no. 4, pp. 579– 641, 2001.

[12] R. Sabourin, G. Genest, and F. Prˆeteux, "Off-line signature verification by local granulometric size distributions," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 9, pp. 976–988, 1997.

[13] C. Quek and R.W. Zhou, "Antiforgery: a novel pseudo-outer product based fuzzy neural network driven signature verification system," Pattern Recognition Letters, vol. 23, no. 14, pp. 1795–1816, 2002.

[14] B. Fang, C. H. Leung, Y. Y. Tang, P.C.K.Kwok, K. W. Tse, and Y. K. Wong, "Off-line signature verification with generated training samples," IEE Proceedings - Vision, Image and Signal Processing, vol. 149, no. 2, pp. 85–90, 2002.

[15] Y.Mizukami,M. Yoshimura, H.Miike, and I. Yoshimura, "An off-line signature verification system using an extracted displacement function," Pattern Recognition Letters, vol. 23, no. 13, pp. 1569–1577, 2002.

[16] H. Baltzakis and N. Papamarkos, "A new signature verification technique based on a two-stage neural network classifier," Engineering Applications of Artificial Intelligence, vol. 14, pp. 95–103, 2001.

[17] A. El-Yacoubi, E. J. R. Justino, R. Sabourin, and F. Bortolozzi, "Off-line signature verification using HMMs and cross-validation," in IEEE International Workshop on Neural Networks for Signal Processing, pp. 859–868, Sydney, Australia, December 2000.

[18] E. J. R. Justino, F. Bortolozzi, and R. Sabourin, "Off-line signature verification using HMM for random, simple and skilled forgeries," in International Conference on Document Analysis and Recognition, vol. 1, pp. 105–110, Seattle, Wash, USA, 2001.

[19] F. Leclerc and R. Plamondon, "Automatic signature verification: the state of the art, 1989–1993," International Journal of Pattern Recognition and Artificial Intelligence, vol. 8, no. 3, pp. 643–660, 1994.

[20] R. Plamondon and G. Lorette, "Automatic signature verification and writer identification—the state of the art," Pattern Recognition, vol. 22, no. 2, pp. 107–131, 1989.

[21] R. Sabourin, R. Plamondon, and G. Lorette, "Off-line identification with handwritten signature images: survey and perspectives," in Structured Document Image Analysis, H. Baird, H. Bunke, and K. Yamamoto, Eds., pp. 219–234, Springer-Verlag, NY, USA, 1992.

[22] J. Gupta and A. McCabe, "A review of dynamic handwritten signature verification," Tech. Rep., James Cook University, Australia, 1997.

[23] R. N. Bracewell, Two-Dimensional Imaging, Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[24] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[25] J. R. Deller Jr., J. H. L. Hansen, and J. G. Proakis, Discrete-Time Processing of Speech Signals, IEEE Press, Piscataway, NJ, USA, 1999.

[26] R. Auckenthaler,M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," Digital Signal Processing, vol. 10, no. 1, pp. 42–54, 2000.

[27] B. Herbst. J. Coetzer. and J. Preez, "Online Signature Verification Using the Discrete Radon Transform and a Hidden Markov Model," EURASIP.Journal on Applied Signal Processing, vol. 4, pp. 559–571, 2004.

[28] D. Lowe, "Distinctive Image features from Scale- invariant Keypoints.," International Journal of Computer Vision., vol. 60, no. 2, pp. 91–110, 2004.

[29] S. I. Abuhaiba, " Offline Signature Verification Using Graph Matching," Turk J Elec Engine, vol. 15, no. 1, 2007.

[30] A. I. Abdullah, " Handwritten Signature Verification Using Image Invariants and Dynamic Features," Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation, 2006.

[31] G. F. Russel. A. Heilper. B. A. Smith. J. Hu. D.Markman. J. E. Graham. T. G. Zimmerman. and C. Drews, " Retail Application of Signature Verification," Proceedings of SPIE 2004, vol. 5404, pp. 206–214, August 2004.

[32] S. Srihari. K. M. Kalera. and A. XU, "Offline Signature Verification and Identification Using Distance Statistics," International Journal of Pattern Recognition And Artificial Intelligence , vol. 18, no. 7, pp. 1339–1360, 2004.

[33] S. Reddy. B. Maghi. and P. Babu, "Novel Features for Offline signature verification.," Journal of Computer,Communication and Control., vol. 1, pp. 17–24, 2006.

[34] B. A. Jesus. A. Migual. and M. Traveiso, " Off-line Geometric Parameters for Automatic Signature Verification Using Fixed Point Arithemetic," IEEE Trans.Pattern Analysis and Machine Intelligence, vol. 27, no. 6, pp. 341–356, June 2005.

[35] K. B. Viyanak, "A color code Algorithm for Signature Recognition," International Journal of Pattern Recognition And Artificial Intelligence, vol. 6, no. 1, pp. 1–12, 2007.

[36] Z. Lin. W. Liang. and R. C. Zhao, "Offline signature verification Incorporating the prior model," International Conference on Machine Learning and Cybernetics, vol. 3, pp. 1602– 1606, 2003.

[37] T. S. enturk. E. ¨ Ozgunduz. and E. Karshgil, " Handwritten Signature Verification Using Image Invariants and Dynamic Features," Proceedings of the 13th European Signal Processing Conference EUSIPCO 2005,Antalya Turkey, 4th-8th September, 2005.

[38] B. C. Lovell. V. K. Madasu. and K. Kubik, "Automatic Handwritten Signature verification system for Australian Passports," Science,Engineering and Technology Summit on Counter-Terrorism,Canberra, pp. 53–66, 2004.

[39] H. S. Srihari and M. Beall, "Signature Verifcation Using Kolmogrov Smirnov Statistic," Proceedings of International Graphonomics Society,Salemo Italy , pp. 152–156, june,2005.

[40] Check fraud statistics, "National fraud centre," http://www.ckfraud.org/statistics.html - Retrieved february 22,2008, 2008.

[41] Embassy of the United States Kampala Uganda, "Business fraud warning," http://kampala.usembassy.gov/business fraud warning2.html - Retrieved february 22,2008, 2008.

[42] Bank of Uganda, "Bankfraud," http://www.bou.or.ug/BANKFRAUD.pdf-Retrieved february 22,2008, 2008.

[43] S. N. Srihari and A. Xu., " Learning Strategies and Classification Methods for Offline Signature Verification," Proceedings of the 7th internationalWorkshop on Frontiers in handwriting recognition, 2004.

[44] F. Bortolozi. E. R. Justino., A. E. Yocoubi. and R. Sabourin, "An Off-line Signature Verification System Using HMM and Graphometric features," DAS 2000,4th IAPR International on Document Analysis Systems,Rio de Jeneiro, 2000.

[45] K. Faez. M. Dehghan. and M. Fathi, "Signature Verification Using Shape Descriptor and Multiple Neural Network," IEEE TENCON 1997-Speech and Image Technologies For Computing and Telecomunications, pp. 415–418, 1997.

[46] H. Hammandlu and V. M. Krishna, " Off-line Signature Verification and Forgery detection using Fuzzy modeling," Pattern Recognition, vol. 38, pp. 341–356, 2005.

[47] M. Blumenstein. S. Armand. and Muthukkumarasamy, "Off-line SignatureVerification using the Enhanced Modified Direction Feature and Neuralbased Classification," International Joint Conference on Neural Networks, 2006.

[48] Q. Qianghua. S. Yaiqian and P. Jingui, "Offline Signature Verification Using Geometric Features Specific to Chinese Handwritting," 24th Int. Conf.Information Technology Interfaces, June 24-27,2002.

[49] Y. Y. Wang. C. H. Leung. Y. Y. Tang. P. C. K. Kwok. K.W. E. Tse. B. Fang and Y. K. Wong, "A Smoothness Index Based Approach for Off-line Signature Verification," Proceedings of the Fifth International Conference on Document Analysis and Recognition , pp. 785–787, September 9,1999.

[50] H. Miike. Y. Mizukami., M. Yoshimura and I. Yoshimura, "An Offline signature verification system using extracted displacement function," Pattern Recognition Letter, vol. 23, no. 13, pp. 1569–1577, 2002.

[51] C. H. Leung. Y. Y. Tang. P. C. K. Kwok. K. W. Tse. B. Fang. and Y. K. Wong, "Off-line signature verification with generated Training samples.," IEEE proceedings Vision,Image and Signal processing., vol. 149, no. 2, pp. 85–90, 2002.

[52] D. Lowe, "Object Recognition from Local Scale Invariant features.," In International Conference on Computer Vision, pp. 1150–1157, 1999.

[53] H. Kim. H. Lee. and H. K. Lee, "Robust Image Watermarking using Local Invariant Features,"Proceedings of SPIE, vol. 45, no. 3, 2006.

[54] G. Enrico. B. Manuele., L. Anderea. and T. Massimo, "On the use of SIFT features for face authentication.," In the proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop, pp. 91–110, 2006.

[55] P. Schwarz, "Recognition of Graffiti," BS Thesis,The University of Western Australia, 2006.

[56] L. Dlagnekov, "Video-based Car Surveillance: Licence plate, Make and Model Recognition," MSc Thesis,University of Calfornia,San Diego, 2005.

[57] P. Sharath. P. UnSang. and A. K. Jain, "Robust Image Watermarking using Local InvariantFeatures," Proceedings of SPIE Defense and Security symposium Orlando,Florida, 2008.

[58] T.F. EL-Maraghi, "Matlab sift tutorial," Available from: ftp://ftp.cs.utoronto.ca/pub/jepson/teaching/vision/2503/SIFTtutorial.zip - Retrieved July 5,2008.

[59] I. H. Witten and E. Franh, Data Mining, Elesevier, 2005.

[60] B. Zhang. S. N. Srihari., C. I. Tomai. and S. J. Lee, "Individuality of Numerals.," Proceedings International Conference on Document Analysis and Recognition (ICDAR) Edinburgh, Scotland, pp. 1096–1100, 2003.

[61] D. de Ridder. F. van der Heijden., R. P.W. Duinn. and D. M. J. Tax, Classification,Parameter Estimation and State Estimation:An Engineering Approach using MATLAB, John Wiley and Sons Ltd, 2004.