Representation of Ballistic Strokes of Handwriting for Recognition and Verification

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Electronics and Communication Engineering

by

Prabhu Teja S
201032003
prabhuteja.s@research.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
January 2015

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Representation of Ballistic Strokes of Handwriting for Recognition and Verification" by Prabhu Teja S, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. Anoop M Namboodiri

To Family and Friends

Acknowledgments

No man's work is entirely to his own credit and so isn't mine. The support and effort of many others resulted in this thesis. First and foremost, I would like to thank my thesis advisor, Dr Anoop Namboodiri. Anoop sir, as I call him, has been not only a great teacher but also a great source of inspiration. He was patient with me when I steered off-track with my research and put me back on track. He is undoubtedly the most patient professor I've ever had. He taught me even the most basic and the most important of aspects of research and more importantly life. The discussions I had with him on various topics ranging from machine learning to cricket to politicians are something that I've learnt from and will treasure for a very long time.

I have a few friends to thank for being a great support system: *Vasu and Gopi*, for all the fun and for bearing my idiosyncrasies! Gopi and I would like to think of ourselves as the light-bearers of the reddit and 4chan using community in our social clique. *Siddu*, for not only being a senior but also a good friend. I'll always remember and cherish the biriyani trips. *Ravi and Arpita* for all the fun and the not so fun discussions about everything; the lively late night chats that used to make my day. Thank you for proving that nothing is impossible!

A lot of people contributed a lot to the life at CVIT. I owe a lot to Anoop sir's group from whom I not only learned a lot but also had a lot of fun with: Koustav, Rohan, Aniket, Vijay, Mihir, Vikram. Friends from CVIT who have made me feel comfortable: Praveen, Nagender, Parikshit, Tripathi. I'll never forget the trips to the cafeteria having the most random of banters.

My parents who have provided me with the best possible for everything I have asked for.

A group that I'll be forever indebted to are the folks at Roshni, for their incredible heart.

Abstract

The primary computing device interfaces are moving away from the traditional keyboard and mouse inputs towards touch and stylus based interactions. To improve their effectiveness, the interfaces for such devices should be made robust, efficient, and intuitive. One of the most natural ways of communication for humans has been through handwriting. Handwriting based interfaces are more practical than keyboards, especially for scripts of Indic and Han family, which have a large number of symbols. Pen based computing serves four functionalities: a. pointing input b. handwriting recognition c. direct manipulation d. gesture recognition. The second and fourth problems fall under the broad umbrella of pattern recognition problems. In this thesis we focus on efficient representations for handwriting.

In the the first part of this thesis we propose a representation for online handwriting based on ballistic strokes. We first propose a technique to segment online handwriting into its constituent ballistic strokes based on the curvature profile. We argue that the proposed method of segmentation is more robust to noise, compared to the traditional speed profile minima based segmentation. We, then, propose to represent the segmented strokes as the arc of a circle. This representation if validated by the Sigma-lognormal theory of handwriting generation. These features are encoded using a bag-of-words representation, which we name bag-of-strokes. This representation is shown to achieve state-of-art accuracies on various datasets.

In the second part, we extend this representation to the problem of signature verification. To define a verification system, a similarity metric is to be defined. We propose a metric learning algorithm based on the Support Vector Machine (SVM) hyperplanes learned to separate the training data. This results in a very simple metric learning strategy that capable of being modified to increase the number of users registered by the verification system. We experiment with this technique on the publicly available SVC-2004 database and show that this method results in accuracies applicable to practical scenarios.

Contents

Chapter Page

1 Introduction . . . 1
1.1 Capture of Online handwriting . . . 2
1.2 Handwriting production . . . 3
1.2.1 Hollerbach's Oscillatory model . . . 4
1.2.2 Delta lognormal model . . . 5
1.2.3 Sigma lognormal model . . . 7
1.2.4 Beta-elliptic model . . . 8
1.3 Problem statement and Organization . . . 9

2 A Ballistic Stroke Representation of Online Handwriting for Recognition . . . 11
2.1 Introduction . . . 11
2.2 Previous work . . . 11
2.2.1 Structural & rule based methods . . . 12
2.2.2 Statistical methods . . . 13
2.2.2.1 Implicit methods . . . 13
2.2.2.2 Markov models . . . 14
2.2.2.3 Prototype methods . . . 14
2.3 Curvature based stroke segmentation . . . 14
2.4 Circle based representation of strokes . . . 16
2.5 Experimental Results and Analysis . . . 18
2.5.1 Malayalam Dataset . . . 18
2.5.2 UJI Penchars . . . 19
2.5.3 Cross-language recognition . . . 19
2.5.4 Data from Capacitive sensing device . . . 19
2.5.5 Random vector quantization . . . 20
2.6 Conclusion . . . 20

3 Stroke Level Representation for Signatures . . . 22
3.1 Introduction . . . 22
3.1.1 Signature biometrics . . . 24
3.1.2 Error metrics in biometrics . . . 24
3.2 Representation of signatures . . . 26
3.3 Distance measure . . . 26
3.4 Results . . . 28
3.4.1 Database . . . 28

3.4.2 Experiments . . . 28
3.4.3 Comparison with other methods . . . 30
3.4.4 User specific thresholds . . . 32
3.5 Conclusion . . . 32

4 Conclusions . . . 33
4.1 Summary . . . 33
4.2 Future work . . . 33

Appendix A: Background . . . 35
A.1 Machine learning paradigms . . . 35
A.2 Support Vector Machines based Classification . . . 35
A.3 K-means Clustering . . . 39
A.4 Bag of words model . . . 40

Bibliography . . . 44

List of Figures

Figure Page

1.1 Illustrations for resistive and capacitive sensing devices. Images from http://techexplainer.wordpress.com/2012/04 vs-capacitive-touchscreen/ 2
1.2 Schematic for magnetic sensor. Image from [47] 3
1.3 The intention to write is transfered to a semantic level, which is further transformed into words (lexical and syntactical level). The individual letters (graphemes) are transformed into shapes (allographs) based on writer characteristics and contextual information. Below this level, the allographs are transformed into basic movements called strokes . . . 4
1.4 Model of a synergy made up of two complex neuromuscular systems where the different components are hierarchically coupled 6
1.5 Elliptical arc representation for a simple stroke 9

2.1 The letter *a* with four ballistic strokes. The velocity profile has seven minima indicating eight strokes, while the proposed curvature profile computation gives three maxima indicating four ballistic strokes. 12
2.2 This flowchart describes the skeleton of method proposed in [3] 13
2.3 The character z segmented at speed minima and curvature maxima. Note that the spurious stroke boundary is removed and the location of boundaries are corrected by the curvature based segmentation. 16
2.4 Example of segmentation and representation 17

3.1 Biometric system error rates.(a) FAR and FRR for a given threshold t are displayed over the genuine and impostor score distributions; (b) The curve relating FAR to FRR at different thresholds is the receiver operating characteristics (ROC). Image from [31] . . 26
3.2 ROC (Receiver operating characteristic) curve for random forgeries for varying number of words in BoS representation. The EERs for each of the plots are 2.5%, 2.50%, 1.97%, 2.04%, 1.67%, 1.43% respectively. Plots are best viewed in colour on a monitor. . . . 28
3.3 ROC (Receiver operating characteristic) curve for skilled forgeries for varying number of words in BoS representation. The EERs for each of the plots are 23.75%, 24.12%, 22.88%, 21.75%, 23.38%, 22.88% respectively. Plots are best viewed in colour on a monitor. 29
3.4 ROC curves for various number of classes used to construct the Metric matrix 31

A.1 Optimal max-margin hyperplane in two-dimensional space. Image from [2] 36
A.2 Without being given the class labels of the data, it is obvious that there are 3 clusters in the data. This is the basis of k-means algorithm 40

A.3 Illustration of training of Bag-of-Words(BoW) model 41
A.4 Illustration of testing with Bag-of-Words(BoW) model 42

List of Tables

Table Page

1.1 The variation of shape of top corners with ψ 5

2.1 A table showing percentage accuracies of recognition of various datasets with several feature extraction techniques 19

2.2 Percentage accuracies of recognition for Vector quantization based on k-means and Random Vector Quantization 20

3.1 Some of the existing biometric modalities 23

3.2 Different types of features for Signatures. This table has been shown only to emphasise that the aspects this chapter predominantly deals with are in the domain of feature extraction and not the design of specialised classifiers. The reader is redirected to [41, 27, 17] for an overall study. 25

3.3 Changes in EERs with change in train-test splits 30

3.4 EERs for random forgeries of various systems 30

3.5 EER for user-specific thresholds 32

Chapter 1

Introduction

Making digital documents completely replace the paper variety in offices has remained a dream for a long while now (The myth of the paperless office [66]). It has never been entirely possible because of several advantages that paper documents avail like simplicity of navigation, ease of annotation, quality of display and the ability to function without any additional infrastructure. This revolution of going paperless has occurred in two major forms. *a)* Conversion of all current paper documents into digital format [49, 24, 5], and *b)* Storing all forms of new data directly in digital format. Among documents that are currently generated, the printed ones start their life in the digital form and hence do not require any additional digitization effort. However, handwritten data needs to be scanned and does not yield easily to machine interpretation and search. An alternative would be to capture the handwriting during its generation [58, 71, 70].

The process of acquisition of handwritten data is either online or offline:

- *Offline:* Scans, photographs or any other static capture of handwriting done after the writing process.

- *Online:* Handwriting data that is acquired at the time of writing using devices that capture and store the pen movement as a temporal sequence.

In this thesis we focus on representation and processing of online handwriting. Online handwriting acquisition has seen a steep rise owing to the popularity of devices that are capable of capturing data in such format like smart phones, tablets, and pen-enabled devices. In addition to being a way of data entry, such touch or pen based interfaces allow natural interactions and improve the usability of the device[1] [2]. Most interactions till the advent of such devices has predominantly been through displays, keyboard and mouse. Keyboard based interactions becomes a bottleneck for non-Latin languages with complex scripts. The physical structure of these devices do not conveniently lend themselves complex scripts such as Chinese and Indic. Point-and-click interactions is also tedious when textual input is needed. For touch and pen-based devices, handwriting is the most natural way of data input, both from the points of view of human usability and hardware requirements.

[1] Microsoft Tablet PC Edition

[2] A brief history of tablet and tablet cases http://www.tablet2cases.com/case-o-pedia/about/history/

1.1 Capture of Online handwriting

The technology that is used to capture handwriting determines the quality of the captured handwriting. There are several major systems of handwriting capture which not only vary in the underlying hardware but also in the physical nature, haptic feedback, cost.

- *Resistive devices:* Resistive sensing devices comprise of two transparent, thin layers which are separated by a small space. The bottom-side of the top layer and top of the bottom layer have a coating of an electricity conducting material. Voltage is applied to one layer and is sensed at the other. When a pen-tip presses the outer layer, that layer bends at the point of contact and connects with the bottom layer and closed circuit is established. By the current sensed at the four corners, the location of contact is determined. Since it is a pressure based activation, these devices have the advantage of being used with fingers, pens, styluses.

- *Capacitive devices:* Capacitive sensing devices use glass coated with a conductor. When a conductor like human skin touches the glass, the electrostatic field i.e., the capacitance changes. This change is sent to a controller to detect the touch. Most touch screen phones and tablets use capacitive sensing technology. These devices need the contact surface to be a conductor and hence do not work with stylus or other non-conducting devices. Also, the accuracy of sensing of these devices is comparitively poorer than that of resistive devices.

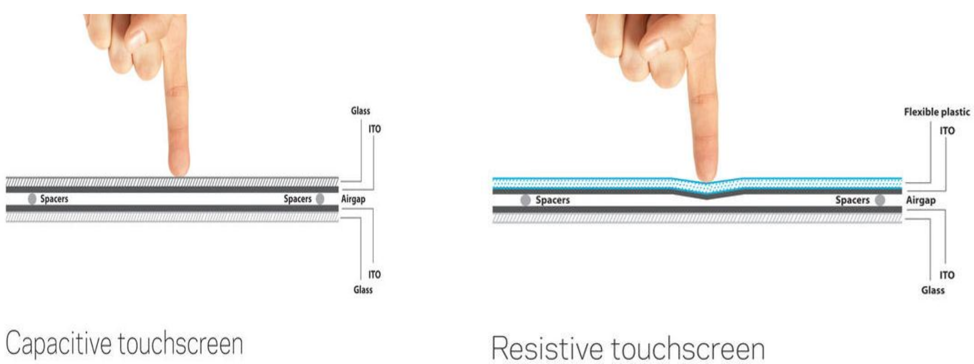


Figure 1.1: Illustrations for resistive and capacitive sensing devices. Images from http://techexplainer.wordpress.com/2012/04/02/resistive-vs-capacitive-touchscreen/

- *Surface Acoustic waves:* Ultrasound waves are passed over the touch screen panel and the disturbances in it are used to sense the touch. Any contaminants like dust interfere with the functioning of the touchscreen.

- *Magnetic tracking:* Most tablet PCs use magnetic tracking technology. These devices have a coils that create a magnetic field which induce a magentic field in the coil in the pen. Coils in the pad pick the field from the pen up and the relative strenghts indicate the pen position. The presence of other magnetic field emitting devices like speakers close to the screen results in degradation of sensing accuracy

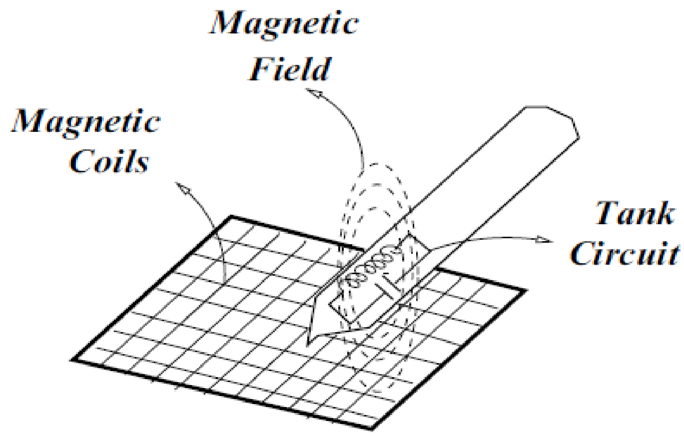


Figure 1.2: Schematic for magnetic sensor. Image from [47]

1.2 Handwriting production

Complex handwriting, say a character or a signature, has been theorized to be produced by hierarchical flow of information through several systems. The exact flow of transformation from an intent to write to the final output of writing is shown in fig1.3. The study of 'strokes' in handwriting has been of great interest to several communities of researchers. Motor control theory posits that strokes are the primitives from which complex movements arise. It is the output of coupled neuromuscular system with one command. In graphonomics literature, it is defined as the trajectory between two consecutive minima in the speed profile of the pen tip. Forensics looks for minor variations in individual stroke patterns for authenticity detection. They have been used to study handwriting deficits in children and for training patients with writers' cramp. They have been used to characterize and diagnose neurodegenrative diseases like Parkinson's, Alzheimer's.[16]

Various models have been proposed to explain and mimic the production characteristics of handwriting. They can be broadly characterized into two types[20]:

a *bottom up approaches,* which mimic the lower level characteristics of handwriting like velocity, acceleration and bear little fidelity towards any neuro-muscular processes eg: [57]

b *top-down approaches,* which focus on psychological aspects like motor learning, movement memory, planning and sequencing, co-articulation and task complexity of strokes, eg: [72]

We now present a few examples of bottom-up models that lead on to the motivations of the thesis.

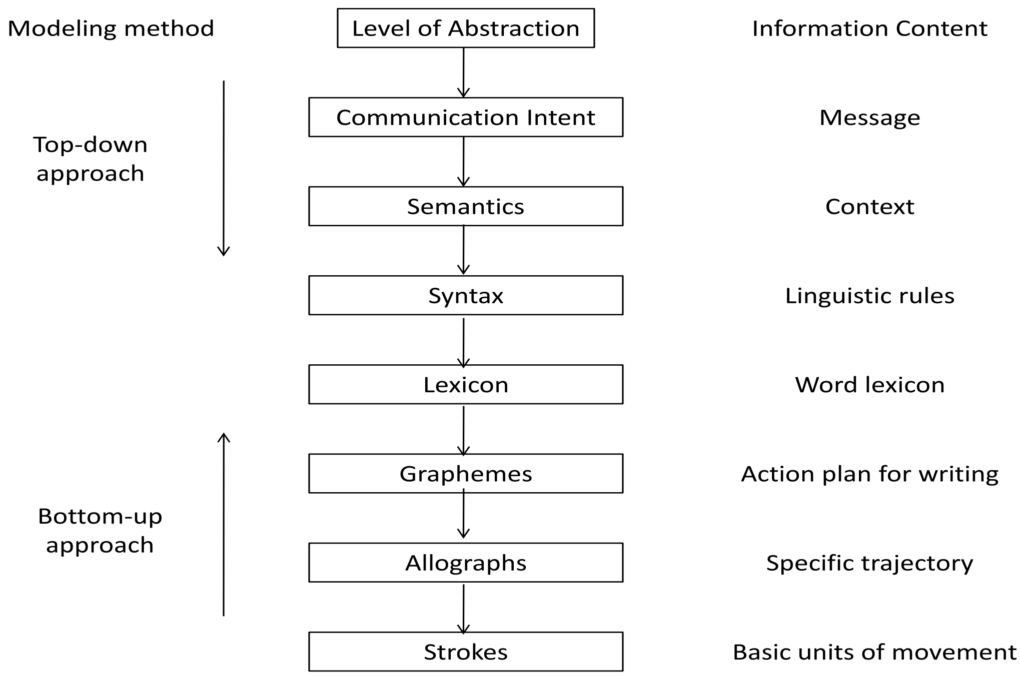


Figure 1.3: The intention to write is transfered to a semantic level, which is further transformed into words (lexical and syntactical level). The individual letters (graphemes) are transformed into shapes (allographs) based on writer characteristics and contextual information. Below this level, the allographs are transformed into basic movements called strokes

1.2.1 Hollerbach's Oscillatory model

One of the first oscillatory models was proposed by John Hollerbach in 1981. In this model, handwriting is seen as the result of two superimposed oscillators in distinct directions[25]. The orthogonal oscillations are responsible for producing letter shape, while the horizontal sweep which is a rightward constant velocity results in spatially separated letters. The modulation of vertical oscillations is used to control the size of the letter and the horizontal oscillations for spatial separation. The oscillators' time evolution is defined as

$$\frac{dx}{dt} = a\sin(\omega_x t + \phi_x) + c \tag{1.1}$$

$$\frac{dy}{dt} = b\sin(\omega_y t + \phi_y) \tag{1.2}$$

where a and b are the horizontal and vertical velocity amplitudes, $\omega_x, \omega_y, \phi_x, \phi_y$ are respectively the frequencies and the phases associated to these directions. c represents the constant magnitude of horizontal sweep. The zero-crossing of vertical velocity is important in a way that controlling horizontal velocity at this point controls the shape of output at corners. Integrating the equations results in a cycloid. All the model parameters are piecewise constant and change between vertical velocity nulls. The value of

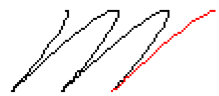
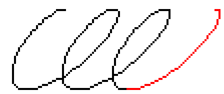
Trace shape	ψ
	1.262
	37.41
	-43.74

Table 1.1: The variation of shape of top corners with ψ

the horizontal velocity at vertical null is given by

$$\psi = \frac{dx}{dt}(t_{y_0}) = c - asin(\phi_x - \phi_y) \tag{1.3}$$

The value of ψ defines the shape of the trace. It it is close to zero, the top corner will be sharp (cusp-like). If it is positive, the corner will be rounded. If it is negative, it will be a full loop.

1.2.2 Delta lognormal model

Kinematic theory[55, 54, 53, 52], proposed by R Plamondon, has been shown to be very successful in reproducing some invariant properties observed in rapid handwriting strokes. This kinematic theory is reliant on the delta-lognormal model. The stroke velocity profile is the output of two neuromuscular systems, one agonist and one antagonist, acting in opposite directions. If systems are assumed to be linear time-invariant, and each of the velocity outputs are $v_{\sigma 1}(t)$,$v_{\sigma 2}(t)$ are given by

$$\begin{aligned} v_{\sigma 1}(t) &= D_1 H_1(t - t_0) \ for \ t \geq t_0 \\ v_{\sigma 2}(t) &= D_2 H_2(t - t_0) \ for \ t \geq t_0 \end{aligned} \tag{1.4}$$

where t_0 is the impulse time, H_1, H_2 are the impulse responses of the agonist and antagonist systems. Thus the global synergy is the difference between the impulse responses of the agonist and antagonist systems, weighted by the respective amplitude of their input commands.

$$v_\sigma(t) = v_{\sigma 1}(t) - v_{\sigma 2}(t)$$

If each of the components has an independent sequence of subsystems then

$$H(t - t_0) = h_1(t - t_0) * h_n(t - t_0) * \cdots * h_n(t - t_0)$$

where * is convolution. From the definition of the central limit theorem (assuming appropriate conditions), $H(t - t_0)$ will tend toward a gaussian curve with a total time delay μ (with respect to t_0) and a response time σ, if the number of subsystems n is sufficiently large. It has been previously shown that the speed profile of a stroke is an asymmetric gaussian curve. Such a coupling of systems does not explain this phenomenon.

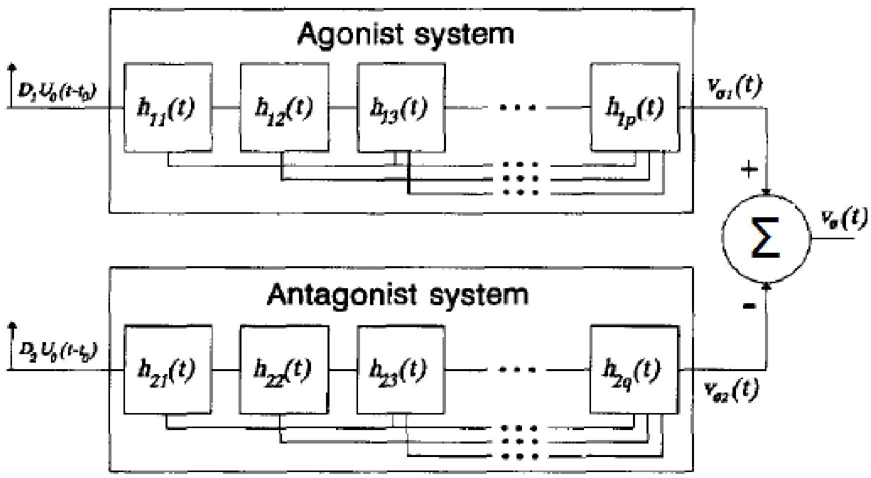


Figure 1.4: Model of a synergy made up of two complex neuromuscular systems where the different components are hierarchically coupled

However, if the sequence of subsystems are hierarchically coupled as shown in fig 1.4, system can be written as $H(e^{t-t_0})$ converging to a gaussian. Here the hierarchiacal coupling means that each of its components is linked to its neighbor as well as connected to a large number of more distant components. Thus the system $H(t - t_0)$ is equal to a lognormal curve. (The mathematical details can be found in [52])

Thus each of these systems(agonist and antagonist) have an impulse response of lognormal function $\Lambda(t; t_0, \mu, \sigma)$. The velocity profile of the handwriting, thus produced can be written as

$$v(t) = \vec{\mathbf{D_1}}\Lambda(t; t_0, \mu_1, \sigma_1) - \vec{\mathbf{D_2}}\Lambda(t; t_0, \mu_2, \sigma_2) \tag{1.5}$$

where

$$\Lambda(t; t_0, \mu, \sigma) = \begin{cases} \frac{1}{\sigma(t - t_0)\sqrt{2\pi}} \exp\left(\frac{[ln(t - t_0) - \mu]^2}{-2\sigma^2}\right) & t > t_0 \\ 0 & elsewhere \end{cases} \tag{1.6}$$

where: D_1, D_2 are input commands corresponding to agonist and antagonist components, t_0 is the time of occurrence of input commands to both agonist and antagonist systems, μ_1, μ_2 are the log-time delays of the neuromuscular systems, σ_1, σ_2 are the log-response times.

1.2.3 Sigma lognormal model

Sigma lognormal model is a generalised model compared to the delta-lognormal model. For very complex movements like signatures, notion of agonist and antagonist systems is very unclear. Thus a single timing parameter of both components (t_0) is no longer required. The sigma-lognormal[50] model posits the output speed of neuromuscular system action is of the shape of a lognormal curve scaled by command parameter (D) and shifted in time by the time of command (t_0), as given in equation1.7. Since the output is considered to be a movement happening along a pivot, the angular position is given by equation 1.8.

$$
\begin{aligned}
|\vec{v}_j(t, P_j)| &= D_j \Lambda(t - t_{0j}; \mu_j, \sigma_j) \\
&= \frac{D_j}{\sigma_j (t - t_0)\sqrt{2\pi}} \exp\left(\frac{[ln(t - t_0) - \mu_j]^2}{-2\sigma_j^2}\right) && (1.7) \\
\phi_j(t; P_j) &= \theta_{sj} + \frac{\theta_{ej} - \theta_{sj}}{D_j} \int_0^t |\vec{v}_j(\tau, P_j)| d\tau \\
&= \theta_{sj} + \frac{\theta_{ej} - \theta_{sj}}{2} \left[1 + erf\left(\frac{ln(t - t_0) - \mu_j}{\sqrt{2}\sigma_j}\right)\right] \text{ , } where && (1.8) \\
P_j &= [D_j,\, t_{0j},\, \mu_j,\, \sigma_j,\, \theta_{ej},\, \theta_{sj}] && (1.9) \\
erf(x) &= \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt && (1.10)
\end{aligned}
$$

Equations 1.7,1.8 describe the system for one stroke. A complex handwriting has several such systems. The total synergy of coupling of several such systems is a vectorial summation of the velocities of the individual systems.

$$
\begin{aligned}
\vec{v}(t) &= \vec{\Sigma}\Lambda(t; P) = \sum_{i=1}^{M} \vec{v_j}(t; P_j) && (1.11) \\
P &= [P_1^t\, P_2^t \ldots P_M^t] && (1.12)
\end{aligned}
$$

The x & y components of the velocity is given as the speed times the cosine and sine of the angle respectively. The coordinates traced is derived by integrating the v_x and v_y respectively. They are given by:

$$v_x(t;P) = \sum_{i=1}^{M} |\vec{v}_j(t,P_j)| \cos(\phi_j(t;P_j)) \tag{1.13}$$

$$v_y(t;P) = \sum_{i=1}^{M} |\vec{v}_j(t,P_j)| \sin(\phi_j(t;P_j)) \tag{1.14}$$

$$\begin{aligned} x(t;P) &= \int_0^t v_x(t;P) \\ &= \sum_{i=1}^{M} \frac{D_j}{\theta_{ej} - \theta_{sj}} [\sin(\phi_j(t;P_j)) - \sin(\theta_{sj})] \end{aligned} \tag{1.15}$$

$$\begin{aligned} y(t;P) &= \int_0^t v_y(t;P) \\ &= \sum_{i=1}^{M} \frac{D_j}{\theta_{ej} - \theta_{sj}} [-\cos(\phi_j(t;P_j)) + \cos(\theta_{sj})] \end{aligned} \tag{1.16}$$

1.2.4 Beta-elliptic model

The Delta-lognormal function puts some constraints on the shape of the profile because it is an unbounded function. To address this particular problem, the beta-elliptic model[8] has been proposed.

The global velocity response of system is is described by a *Beta function* $\beta(t, t_0, t_1, p, q)$ where t_0 is the starting time and t_1 is the ending time of the synergy.

$$\beta\left(t, t_0, t_1, p, q\right) = \left(\frac{t_1 - t}{t_1 - t_c}\right)^p \left(\frac{t - t_c}{t_c - t_0}\right)^q \tag{1.17}$$

$$t_c = \frac{p \times t_1 + q \times t_0}{p + q} \tag{1.18}$$

where t_c is the instant at which the velocity reaches its maximum. Compared to the Delta-lognormal model, beta model leads to a velocity profile description that is more flexible, and therefore that can fit better the diversity that might arise in hand-writing data.

Apart from these four parameters of the Beta function, each elementary *stroke* is also characterized in the spatial domain by an ellipse. Each ellipse can be described with five parameters: a and b are respectively the semi-major axis and semi-minor axis of ellipse, (x_0, y_0) is the center of ellipse. The angle defines the angle of the semimajor axis from the x-axis. From two points (A, B) of one stroke, the parameters (θ, a, b) are calculated. These points A and B correspond respectively to the minimum and the maximum values of the velocity profile.

$$\frac{(X - x_0)^2}{a^2} + \frac{(Y - y_0)^2}{b^2} = 1 \tag{1.19}$$

where $\begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix}$

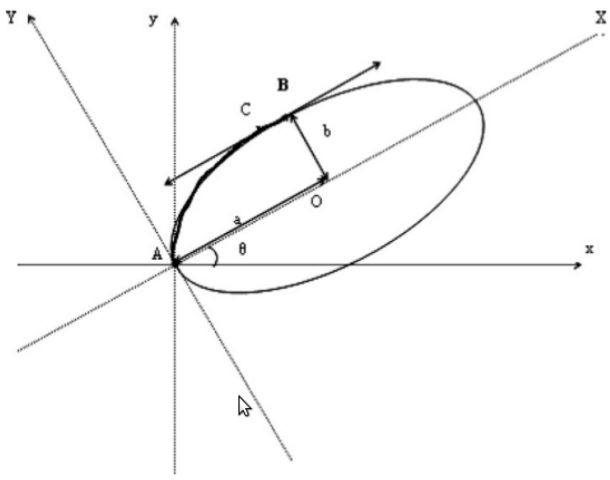


Figure 1.5: Elliptical arc representation for a simple stroke

Thus a stroke is represented in the space and velocity domains by a curvilinear velocity starting at time t_0 at an initial point and moving along an elliptic path. It is to be noted at the curvature of an ellipse is not constant. The representation by the ellipse is shown in fig:1.5. It is obvious that a stroke is characterized by nine parameters; the first four parameters reflect the global timing properties and velocity information, and the last five parameters describe the global geometric properties and spatial domain information. The Beta-elliptic model considers a simple movement as the response of the neuromuscular system, which is described by an elliptic trajectory and a Beta velocity profile. The major difference between the beta-elliptic model and the lognormal model is that it considers each stroke to be of variable curvature, whereas the lognormal model considers constant curvature. While the beta-elliptic model is more generalised and can accommodate more variations than the lognormal model, it is unclear yet if these additional mathematical advantages have any bearing on the description of the handwriting i.e., it is unknown if that generalisation can actually help model handwriting better.

1.3 Problem statement and Organization

One of the major steps in a pattern recognition system is the representation of the input signal. The common and effective ways of representing handwriting till now have been using resampling techniques (equi-spaced, equi-time, random) or some local representations in terms of change of angles between subsequent samples[58]. None of these representations have any theoretical justifications for their functioning. In section1.2, we presented a few widely known methods that explain how handwriting is produced. Very little work has been done in incorporating these handwriting production models in building systems [35, 34]. This thesis is a step towards using the production characteristics of handwriting towards recognition and verification tasks.

This thesis is organized as follows.

- Chapter 2 introduces the ideas of online handwriting recognition and presents an overview of the existing techniques. Then an argument is put forth for the segmentation of a handwriting trace into its constituent strokes is more reliable in the spatial domain than in the temporal domain. An algorithm is devised that uses the constituent strokes of a given character to recognize it.
- Chapter 3 introduces the concept of signature biometrics. An algorithm similar to that of the algorithm proposed for handwriting recognition is proposed. A metric learning strategy is proposed to learn the similarity metric and shown to achieve practically viable accuracy.
- Concluding remarks and future directions are presented in Chapter 4
- Background for some of the important theoretical concepts is provided in Appendix A.

Chapter 2

A Ballistic Stroke Representation of Online Handwriting for Recognition

2.1 Introduction

As argued previously, handwriting based interfaces are the way ahead for modern UI designs. Unfortunately the sensitivity and robustness of most touch and even pen based interfaces makes handwriting recognition difficult when compared to dedicated pen interfaces for Tablet PCs or stand alone digitizers.

The error in sensing the pen or finger tip, both in spatial and temporal dimensions has adverse affects on any recognition algorithm. Consider the problem of detecting the ballistic strokes from pen or finger movement in a mobile device. The most common approach is to detect minima in pen velocity as boundaries of ballistic strokes. However, as Figure 2.1 demonstrates, the noise introduced in such devices would make the process extremely difficult. The error in digitizing devices may be categorized into two: spatial and temporal. Spatial error is introduced due of limits of spatial resolution of the digitizer and noise within the sensing hardware. The temporal sampling error is caused by delays in acquiring a sample due when the recording hardware is busy, and is especially detrimental to velocity based stroke detection as seen from Figure 2.1 or in the computation of dynamic features for recognition.

In this chapter, we develop a robust method for segmentation of ballistic strokes. The detected strokes can be used for recognition of handwritten characters as well as other applications. We also propose a fixed length representation of a trace from the constituent ballistic strokes, which we refer to as Bag-of-strokes as explained later. Through the rest of the chapter, we refer to as trace, the handwriting that has linguistic sense and correctness. The terms ballistic stroke and stroke are used interchangeably.

2.2 Previous work

The major issues that a recognizer has to cope with are of four types [65]:

- *Geometric variations:* Changes that occur in factors like position and size.
- *Allographic variations:* Variations introduced by the writing styles of various writers.

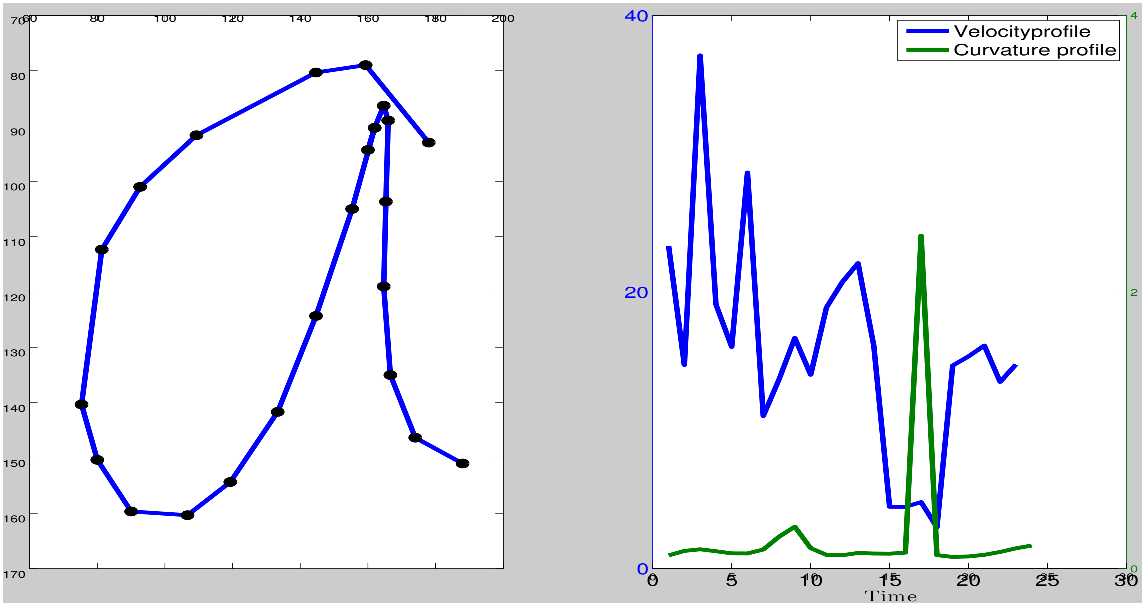


Figure 2.1: The letter *a* with four ballistic strokes. The velocity profile has seven minima indicating eight strokes, while the proposed curvature profile computation gives three maxima indicating four ballistic strokes.

- *Neurophysical & biomechanical factors:* Modification of action plan of writing during execution i.e., change in intended output while writing.

- *Order of production:* Post hoc strokes, delayed strokes.

The efforts in online handwriting recognition can be classified into two major categories: *a.* Structural & rule based methods and *b.* Statistical classification methods.

2.2.1 Structural & rule based methods

It is supposed that character shapes can be described in an abstract fashion without much interest in describing minor variations that creep in during the execution of writing. The fundamental problem with this approach is the difficulty in formulating the rules and automating the generation of rules from large databases. Post 1960s there has been very little work[51], which uses fuzzy rules and grammars that use frequency statistics of some features. The advantage with these methods, as is with rule based methods, is that they require very little training data, except that construction of these rules is a tough task.

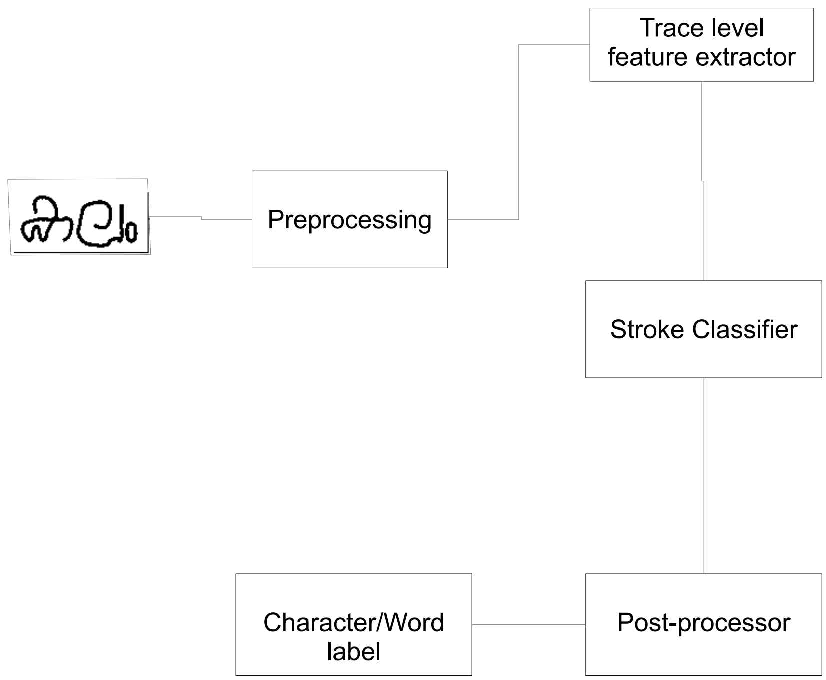


Figure 2.2: This flowchart describes the skeleton of method proposed in [3]

2.2.2 Statistical methods

In statistical methods, a shape is described by a fixed number of features describing some characteristics so that the feature space is a multidimensional probability distribution. The major classification of these methods are detailed below.

2.2.2.1 Implicit methods

Implicit methods generally refer to neural network based methods that rely on statistical characterisation of training data. The major methods are Kohonen's Self-Organized Maps (SOM)[42] and time-delay neural networks (TDNN)[21, 64, 22]. SOMs use vector quantization (like k-means) to map the shapes to possible linguistic interpretations. Convolutional TDNNs use a convolutional architecture like Convolutional Neural Nets(CNN) and use shared network weights to achieve spatial invariance. TDNN also use contextual information by using the data from the previous characters. In some cases, the outputs of the intermediate layers of these neural networks have been used as a feature representation which are used with another classifier like SVMs[23, 59]. But, these networks are tough to train owing to search in the large parameter space and requirement of large training set.

2.2.2.2 Markov models

Markov modeling has long been the forte for recognition in various domains, especially in speech processing and has been adapted to work for online handwriting recognition[26, 68]. Hidden Markov Models(HMM)[60] is a doubly stochastic process *i.e.,* it has an underlying hidden process that influences an observable process. The underlying process has a characterizable state transition probability. The observable handwriting process is characterized by an emission distribution that depends on current state. The observation probabilities for each state can be a discrete probability distribution or it may be modeled as the mixture of continuous probability densities [7]. These methods have predominantly been applied to Latin scripts where dealing with cursive handwriting is of foremost importance, but haven't been able to achieve practical accuracies[28].

2.2.2.3 Prototype methods

In prototype methods, recognition is done by distance based matching with a subset of training set called the prototype set. The test sample is labeled as belonging to the class of most similar prototype. The three important factors of prototype based matching are: selection of prototypes, representation of samples and the definition of appropriate distance measure. The selection of prototypes is pivotal and overall recognition accuracy is dependent on the this. This method falters when the training set doesn't sufficiently represent the character space. The major advantage with this method is that it allows for representation of characters using variable length features to be matched using Dynamic Time Warping (DTW)[63] and has been shown to achieve reliable accuracies.

The implicit and the Markovian methods generally operate at word-level recognition and are generally constrained to the lexicon on which the network is trained. A more general architecture is shown in 2.2 in which each of the symbols are individually recognized and then the word is completely recognized by using a higher level language model that checks for grammatical and syntactical correctness.

2.3 Curvature based stroke segmentation

All of the previous work assume that the data obtained from the digitizer is clean and can be segmented easily using the velocity profile of the pen/finger trace. With increasing use of mobile devices for recognition, there is a trade-off in the quality of the data being dealt with. Input data to PDAs and tablets are affected by sensor limitations, environmental conditions such as humidity,temperature and EMI from various components. Hence the problem of robust segmentation of ballistic strokes is complementary to the existing works on modeling of hand movement, where segmented strokes are used as basis to estimate the model. We now look into the details of the segmentation process in Section 2.3, before developing a representation for traces suitable for recognition (see Section 2.4).

Traditional method for ballistic stroke segmentation divides a trace at instants of time when speed profile reaches a local minimum [10]. While this method is ideal for non-noisy data, it falters in presence of noise in data capture. There are two major types of distortions induced during data capture. The spatial error described in section 2.1 can be mitigated to a large extent by smoothing and parametric curve fitting such as splines due to the smooth nature of hand movements [48]. Due to temporal noise distortions calculation of velocity becomes erroneous. Most devices specify a temporal frequency of data capture. However, the actual instances of recording of pen tip usually varies slightly from it and the interval between such recordings does not remain constant.

In online handwriting, the trace of a pen or finger is recorded as the sequence: $(x_t, y_t), 0 \leq t \leq N$. The instantaneous acceleration of the pen tip is proportional to the vectorial sum of the force applied by the hand and the frictional force. Since the force applied on the pen by the fingers is finite and bounded, the third derivatives functions x_t & y_t should exist. Based on these, we consider cubic splines to represent and model handwriting. Specifically, we choose a basis spline representation, where the order of the basis function is chosen to be 3. The two parameters that determine the behavior of the spline fit are: a) order of basis b) knot point positions. The lowest order spline that satisfies the smoothness constraint is 3, as argued previously. The choice of knot vectors defines the fit of the curve to the data. A uniform knot vector sequence i.e., equally spaced knots, is used, so as to make sure that all parts of the trace are given equal importance. A farther enough spacing of points gives the required robustness to noise, as shown in [48]

Splines impose the smoothness constraint, which minimizes the integral of the squared second derivative along the curve [15], i.e.,

$$\int S''(t)^2 dt \leq \int G''(t)^2 dt \tag{2.1}$$

where $S(t)$ is the spline function, $G(t)$ is the set of all functions that have continuous second derivatives.

The *two-thirds power law* [39] states an inverse cubic relationship between tangential hand-speed and curvature of trajectory of motion. This inverse relationship allows us to use the maxima in curvature of a trace to segment the ballistic strokes instead of minima of velocity.

Curvature κ at point on a curve, is the measure of rate of change of the tangent at that point to neighboring points. In arc length parameterization, curvature may be defined as [37] the rate of change of tangential angle with arc length.

$$\kappa = \frac{d\phi}{ds} \tag{2.2}$$

After spline fitting to handwriting signals, it can be assumed that there is no spatial sensing noise that deviates from the handwriting curve. Spline evaluation at integral units of time can be understood as dropping the noisy point to the point closest on the spline curve. Ideally, that might not be the correct point given that time instant. In other words, now there is a deviation from the ideal point along the spline curve. This cannot be traditionally termed as error in spatial sampling but as an error in time sampling i.e., sampling not being done at uniform intervals of time. This is evident in figure2.3. There is a clear difference between segmenting the strokes at speed minima and curvature maxima.

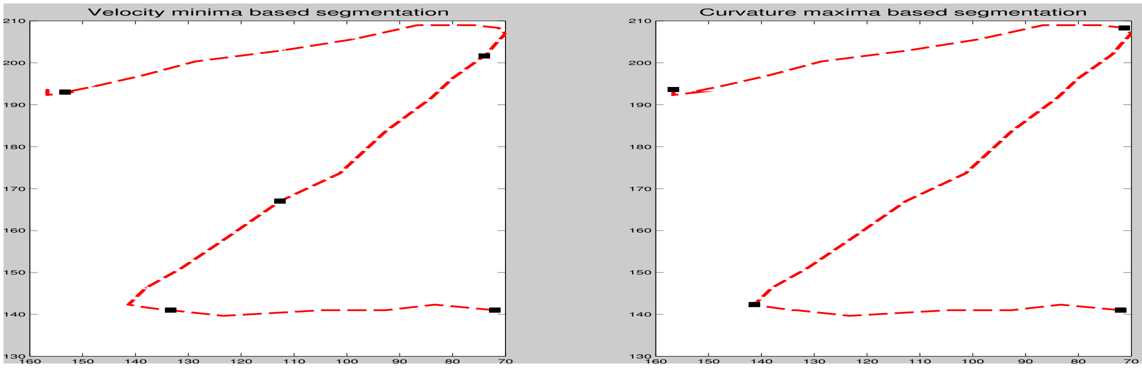


Figure 2.3: The character z segmented at speed minima and curvature maxima. Note that the spurious stroke boundary is removed and the location of boundaries are corrected by the curvature based segmentation.

Lemma 2.3.1 *If the cubic spline fit of the handwriting trace retains only spatial errors along the curve in addition to temporal errors, the computed curvature values and hence the ballistic stroke segmentation based on it is independent of spatial and temporal noise.*

Proof As noted before, any error in the spatial location of the pen tip is indistinguishable from an error in the time instant of sampling of the curve. Hence we can assume that after the spline fit, we have a curve with correct spatial trajectory, which is probably erroneous in the temporal points of sampling. If we use Equation 2.2, which is independent of time as the definition of curvature, the resulting value of curvature would be accurate.

2.4 Circle based representation of strokes

From the sigma log-normal model, a ballistic stroke, spatially, can be described as a pivotal movement of the hand along the arc of a circle [50]. Each basic unit i.e., a stroke is hypothesized to be an arc of a circle. In order to spatially characterize a stroke, the center, radius of the circle, the starting and ending points of the arc are required. To each of the strokes calculated from the curvature profile, a circle is fit by minimizing the sum of squared radial deviations.

$$\min_{x_0, y_0, r} \sum_{i}^{m} x_i^2 + y_i^2 - 2x_0 x_i - 2y_0 y_i + x_0^2 + y_0^2 + r^2$$

Let $-2x_0 = a_1, -2y_0 = a_2, x_0^2 + y_0^2 + r^2 = a_3$. For $i = 1 \ldots m$, the above equation can be represented in matrix form as,

$$\left(X \quad Y \quad \mathbf{1} \right) \left(a_1 \quad a_2 \quad a_3 \right)^T = - \left(X \circ X + Y \circ Y \right) \tag{2.3}$$

where $A \circ B$ denotes the Hadamard product of two matrices A & B, X is a column vector of x co-ordinates and Y is a column vector of y co-ordinates of the points to which a circle has to be fit, $\mathbf{1}$ is a column vector of 1 of length m. The least-squares fit to the system of 2.3 can be solved by calculating the Moore-Penrose pseudoinverse. The center and the radius of the circle can be computed from 2.3 by

$$x_0 = -\frac{a_1}{2}, y_0 = -\frac{a_2}{2}, r = \sqrt{\frac{a_1^2 + a_2^2}{4} - a_3}.$$

An equivalent of describing the circular arc with the starting and ending points is taking the angles of the lines joining the center of the circle and two ends of the stroke with x-axis. Thus the representation $(x_0, y_0, r, \theta_s, \theta_e)$ describes completely a stroke in space, where θ_s, θ_e denote the angles at the starting and ending points of the stroke.

Each trace is normalized in space to fit into a unit square. The sensitivity of the circle parameters (r, x_0, y_0) to curvature of the arc being fit is high. *i.e.*, the values of (r, x_0, y_0) become exponentially large as the the arc approaches a straight line. In order to mitigate this problem, a monotonic function (say hyperbolic tangent) is used to map the values of r from $\mathbb{R}$ to $[0, 1]$. Also, it is intuitive that the variation of the mean of a stroke is less compared to the variations in the center of the fitted circle. So we substitute the mean coordinates of the ballistic stroke in the place of center (x_0, y_0), which we will continue to denote the mean coordinate rather than the center of the fitted circle. The hyperbolic function $\tanh$ is applied only to the radius dimension, as it is the only unbounded ordinate in the representation. Thus the problem that arise during fitting a circle to a straight line is mitigated by using the center of the line and the radius will only tend to 1.

To apply the above model for character recognition, for all the characters in the training set, the strokes are found out and their circle fits are calculated. The circular features *i.e.*, $(x_0, y_0, r, \theta_s, \theta_e)$ of all the strokes are clustered using k-means into k clusters and the centroids of the clusters are stored. Each character is represented as a Bag-of-Words (BoW) model, with the words defined as these centroids.

The feature representation for each character is a k-bit feature vector corresponding to k centroids, which is initialized to zero. For each character's strokes, the circular representation is found. The distance to each of the centroids is calculated and the bit corresponding to the centroid that is closest to the given stroke is set. Thus the entire character is encoded as a string of k-bits. Our representation differs from the traditional Bag-of-Words in such a way that indicator functionality is described rather than a histogram, and we refer to the representation as Bag-of-Strokes. An example of this process for the Malayalam vowel symbol *au* is illustrated in 2.4.

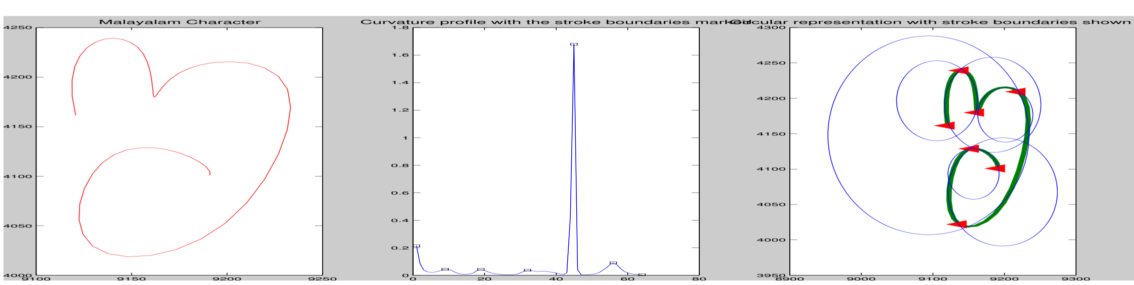


Figure 2.4: Example of segmentation and representation

2.5 Experimental Results and Analysis

We report results on three different representations of the traces. The popular traditional representations are equitime samples and equidistant samples [33]. It has been shown, experimentally, that equidistant resampling has a better discriminative feature space than equitime resampling [3][33]. One of the problems of using equidistant resampling for recognition of Indic languages is that it, on certain occasions, minor discriminative details such as Cusps *vs* small loops are lost. Most of these smaller details happen at areas of high curvature. Thus we also experiment with an alternative method for sampling that is based on curvature.

The curvature is calculated from the spline representation as

$$\kappa(t) = \left| \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\sqrt{\left(\dot{x}^2 + \dot{y}^2\right)^3}} \right|, \tag{2.4}$$

where $x(t)$ & $y(t)$ are the x and y trajectories with time. The cumulative curvature is calculated

$$h(t) = \int_0^t \kappa(t)\, dt$$

The inverse of $h(t)$ is computed and is evaluated at linearly spaced points in the range $\left[0, max\{h(t)\}\right]$. This gives the time-intervals $\{t_i\}$ at which the handwriting has to be resampled. The original handwriting is then interpolated at $\{t_i\}$ to get a resampled version in accordance with the curvature of handwriting. We found that, curvature weighted sampling when clubbed with equidistant resampling performs better than either of them individually (See Table 2.1).

2.5.1 Malayalam Dataset

Malayalam is a Dravidian language that is predominantly spoken in the state of Kerala. We have collected a significant amount of natural handwritten data for Malayalam. Modern Malayalam script has 13 vowels, 36 consonants, and 5 half-consonants. In addition to these, there are several symbols for multiple consonant combinations. We have included the ones currently used in our dataset, ignoring the ones that have fallen out of usance. Unique traces in the data were identified and the complete dataset was labelled accordingly. Our Malayalam dataset contains 106 different traces or classes to be identified, many of them are very similar in shape to others. The actual data was collected as a set of words that were chosen to cover all the trace classes and the set of words were written by over 100 writers. After removal of characters that were incorrectly written, we had 8966 traces in our final dataset. The data was collected using Genius G-Note 7000 digital ink pad.

The optimal number of clusters for Bag-of-Strokes (BoS) was experimentally determined to be $k = 150$. Using the BoS representation, we trained a Support-Vector machine classifier (LIBSVM [12] implementation) with RBF kernel ($\sigma = 0.2$). All experimental results are reported as averages in 3-fold cross validation trials. We are able to achieve an accuracy of 94.55% with BoS representation

	Equidistant Sampling(ED)	Curvature weighted sampling(CS)	ED+CS	Bag of Strokes representation	ED+CS+BoS
Malayalam	84.40	81.75	85.76	94.55	97.75
UJIPenchars	82.51	76.05	86.70	95.8	96.5
Touch Screen data	95	94.5	95.58	93.9	96.2

Table 2.1: A table showing percentage accuracies of recognition of various datasets with several feature extraction techniques

as compared to 85.75% using the representation proposed by Arora *et al.* [3]. Concatenating the equidistant resampling and the BoS representation results in a further improvement in accuracy (97.7%) with a linear kernel in SVM, showing that this method adds information complementary to the one given by the resampling.

2.5.2 UJI Penchars

To test the proposed stroke segmentation and classification, a lower case character subset of publicly available UJIPenchars2 [43] is used. The classification task is of 26 classes. Each class on an average has 120 samples and total number of samples used is about 3116, with some of them being removed as they were not recognizable as corresponding to the class specified.

Using the Bag-of-Words representation, we trained a Support Vector Machine (SVM) classifier. We used the freely available software LIBSVM [12] for classifier design. We performed 4-fold cross validation test on the data. We achieved a peak accuracy of 95.8% using 140 words and RBF kernel with $\sigma = 0.2$, which is higher than [11] method of using approximate DTW with multi-layer perceptron who report an accuracy of 91.8%.

2.5.3 Cross-language recognition

The ballistic stroke is considered to be the most basic unit of handwriting. Any complex movement can be broken down into a set of ballistic strokes. These basic units can be comprehended to be language independent. In-order to test this hypothesis, we used the clusters that were generated in the optimum case of Malayalam data and used those to define the words for the UJIPenchars dataset. The resultant accuracy of such a representation is 95.0%. The apparent 1% decrease in the percentage accuracy is because of the shape characteristics of Malayalam which has lot more complex structures than English does and hence the distribution of centroids is affected by that.

2.5.4 Data from Capacitive sensing device

Capacitive sensors are inherently more noisy and less accurate than resistive sensing devices technology [6]. The various factors that affect the accuracy of sensing are input current noise of amplifier circuits, shot noise effect, excess noise of semiconductor junctions, thermal in resistors apart from the electromagnetic interference. In-order to verify the robustness of our algorithm, recognition experiments

	k-Means	RVQ
Malayalam	94.55	90.03

Table 2.2: Percentage accuracies of recognition for Vector quantization based on k-means and Random Vector Quantization

were ran on small handwriting dataset collected from Google Nexus 7 tablet [1] and a Samsung Galaxy SII mobile phone. 5 participants took part in the data collection for 26 lower case English alphabets, with each of the participants writing each character at-least 10 times. The total number of characters in the database is 1380, giving an average of 53 samples per class.

As seen from Table 2.1, the proposed method performs comparably to equidistant resampling, and provides an improvement over velocity based stroke segmentation, which gives an accuracy of 91.9% on the same dataset (compared to 93.9%) . The result shows the robustness of the proposed segmentation approach to noise in the data. Moreover, the information in the representation complements resampling based methods and the combined accuracy is even higher.

2.5.5 Random vector quantization

To better understand the stroke space i.e., $(x_0, y_0, r, \theta_s, \theta_e)$, we ran an experiment to check the distribution of strokes in the space. Vector quantization has always relied on computation of centroids or some variant of it. More recently, the concept of random vectors to act as the centroids has been studied in the communication systems literature[4].

Random vector quantization (RVQ) is a codebook/ centroid design approach to codebook design that generates the vectors independently from a uniform distribution between the acceptable values of the stroke space. Previous attempts in using this involved the generation of random vectors such that they lie in the complex unit sphere. This has been used for feedback beamforming in multiple antenna systems[62, 4].

For the current problem, the acceptable range of values for $x_0, y_0 \in [0, 1]$, $r \in (0, 1)$,$\theta_s, \theta_e \in [-\pi, +\pi]$. The experiment has been run on the Malayalam dataset. Table 2.2 gives the comparison of percentage accuracies of the various quantization techniques

2.6 Conclusion

In this chapter, we presented an alternative method for ballistic stroke segmentation that uses curvature at any given point by using a parametric representation of handwriting. We observed that this method of stroke segmentation is more robust to sensing and digitizing noise. We proposed a circle based representation using the bag of words technique, where the entire character was represented as a bit vector. The resulting representation is both efficient and effective for character recognition on mobile devices with limited processing and memory capacity. We used the UJIPenchars dataset and a large-class Malayalam dataset to evaluate our algorithm and to show its satisfactory performance.

Chapter 3

Stroke Level Representation for Signatures

3.1 Introduction

Recognizing and ensuring the identity of people is fundamental to several activities in the society. Traditional methods of authentication have been [45]

1 *Possessions* e.g.: Keys, smart cards

2 *Knowledge* e.g.: passwords and PINs

3 *Biometrics* e.g.: any of the traits listed in table 3.1

Systems built on possessions or knowledge can easily falter or can be bypassed with ease [31]. Biometrics refers to identifying an individual based on his or her distinguishing characteristics[9]. The use of biometrics alleviates the problem by confirming or establishing identity of a person by 'who he/she is' rather than 'what he/she knows/possesses'. Thus biometrics does not have the same pitfalls of token or knowledge based approaches in which attributes can be stolen, lost or forgotten. The distinguishing characteristics can be either based on physiological or behavioral traits. Physiological traits are those physical characteristics measured at some point of time, whereas behavioral biometrics are measurements of some action that is carried over an extended period of time. Behavioral biometrics vary with various factors such as state of mind to deliberate alteration.

For the design of a biometric security system, first a biometric trait or modality has to be chosen. For a trait to be successfully used as a biometric trait, it has to be

- *universal* i.e., most people should possess it,
- *discriminative* i.e., two people's traits should be different enough in some domain,
- *permanence* i.e., it should be relative stable over time, and
- *collectability* i.e., users should willing participate in the enrollment phase.

Physiological	**Behavioral**
Face	Signature
Fingerprint	Voice
Hand geometry	Gait
Iris	Keystroke
DNA	Lip motion
Ear Shape	
Odor	
Retina	
Peri-ocular	

Table 3.1: Some of the existing biometric modalities

Several other factors like *measurability*, *performance* have also been deemed important [31][13].

A biometric system can operate in two modes [30]

- Verification - system *verifies* the claim on identity made. It is equivalent to answering the question "Are you who you say you are?". Given an input vector X_Q that in someway encodes the properties of the person to be authenticated and an identity claim I, the problem is to decide whether the claim on identity is true (class ω_1) or false (class ω_2). Usually, X_Q is compared to X_I (registered templates in the system of person I) and the similarity between them should exceed a predefined threshold t.

$$(I, X_Q) \in \begin{cases} \omega_1 & \text{if } S(X_Q, X_I) \geq t. \\ \omega_2, & \text{otherwise.} \end{cases}$$

  The similarity measure is a distance measure that is generally handcrafted for the task at hand. Biometric measurements of the same individual taken at different times are almost never exactly the same. Hence a non-zero threshold t is used.

- Identification - functioning in this mode tries to answer the question of whether a person is known to the system or not. It is equivalent to answering the question of "Whose biometric data is this?". Given an input vector X_Q that in someway encodes the properties of the person to be authenticated, the identity is to be determined I_k, where $k \in \{1, 2 \ldots N\}$.

$$X_Q \in I_k \;\; k = \underset{k}{\operatorname{argmax}}\{S(X_Q, X_{Ik})\} \;\; \forall k = \{1, 2 \ldots N\},$$

  where X_{Ik} is the biometric template corresponding to identity I.

Handwritten signatures, an important behavioral biometric modality, for long have been accepted as as an important discriminative trait. It has been in use even prior to computers and has been used widely for authentication of cheques and other banking transactions. With the massive growth of devices with digital pen inputs like PDAs and touch-screen computers, it is one of the most used biometric traits for small scale systems. The first work on signature verification was published in 1965 by AJ Mauceri [44]. Several survey papers summarizing the state-of-art in this domain have been written [56, 40, 27] detailing various aspects of the system. They focus on both online and offline recognition systems.

Several important traits have been listed before as requirements of a biometric modality. Permanence of a signature is very questionable as a person can alter his/her signature at anytime. Also, the parameters used to distinguish two signatures have never been proven to correspond to implicit, inherent attributes of writer.

3.1.1 Signature biometrics

Signature verification systems, generally, follow the generic pattern recognition pipeline. It has with a preprocessing module, feature extraction module, and a classifier. In addition to these generic steps listed above, there is an identity claim being made before the signature is input to the system. There is also a modification in the classifier module in the pipeline. In a verification system, the query signature template is matched with some or all of the signatures of the identity claimed in the database and a match score is derived, which is thresholded to decide acceptance or rejection of the identity claim.

Modern online handwriting acquisition systems that input elevation and azimuth have also been successfully incorporated into verification. These inputs are processed before being fed to the feature extraction module to remove digitization noises and electronic interferences in the digitizer by techniques like Gaussian smoothing[29], Fourier transformation[32].

Feature extraction, for the task of signature verification, is classified into two major types, based on the information used in the signature data.

- Global or feature based, which represent the entire signature as a vector, based on global trajectory properties.
- Local or functional, based on local properties of the signature.

An exhaustive list of standard features proposed in literature can be found in [17, 27]. A few of them have been listed in table 3.2

3.1.2 Error metrics in biometrics

Any given two samples of a biometric characteristic or trait from the same individual are not exactly the same. Hence the matching score $S(X_Q, X_I)$ quantifies the similarity between input and the template representations i.e., higher the score better the chance of belonging to the same class. The distribution of scores generated from pairs of samples from the same person is called the *genuine distribution* and from different persons is called the *impostor distribution*.

Function based	**Feature based**
Position (x,y) including resampled versions	Total duration of Signature
Velocity v_x,v_y,v	Number of Pen-ups
Acceleration a_x,a_y,a	Statistics of position, velocity & acceleration like Average, Root-mean square, maximum, minimum
Pressure & force (capture device dependent)	Durations of positive velocity & acceleration
Pen inclination (capture device dependent)	Transform based (Fourier,Cosine, Wavelet)
Angle with x-axis	Statisical moments of position, velocity, acceleration
	Directional histograms

Table 3.2: Different types of features for Signatures. This table has been shown only to emphasise that the aspects this chapter predominantly deals with are in the domain of feature extraction and not the design of specialised classifiers. The reader is redirected to [41, 27, 17] for an overall study.

The two mistakes that a biometric system can make are 1) two different persons are mistaken to be the same *(false match/accept)* 2) two templates from the same person are mistaken to be from different people *(false non-match/reject)*[30]. There is always a trade-off between the false acceptance rate (FAR) and the false reject rate (FRR). If the threshold t is increased, FAR decreases and FRR increases; if t is decreased, FAR increases and FRR decreases. The system performance at all operating points (all threshold t) can be analysed using an *receiver-operating characteristic* curve (ROC)3.1. This is, in general usage, the plot between *FAR* and *1-FRR*, which is termed as *Genuine Acceptance Rate (GAR)*. Another important metric, used to measure the general error performance is the *Equal Error Rate (EER)*. It is the error rate of the system when both *FAR* & *FRR* are equal.

In the following section 3.2, we propose a new representation for online signatures , which is based on the representation for handwriting recognition proposed in section 2.4. Feature representations that are used for classification are not always robust for the case of verification and thus require a metric to be learned. So, we propose a metric learning method in section 3.3 that is similar to a method used by the computer vision community called *Random Ensemble METRIC (REMETRIC)*[36]. This has also been extended to fast handwritten stroke retrieval in [67].

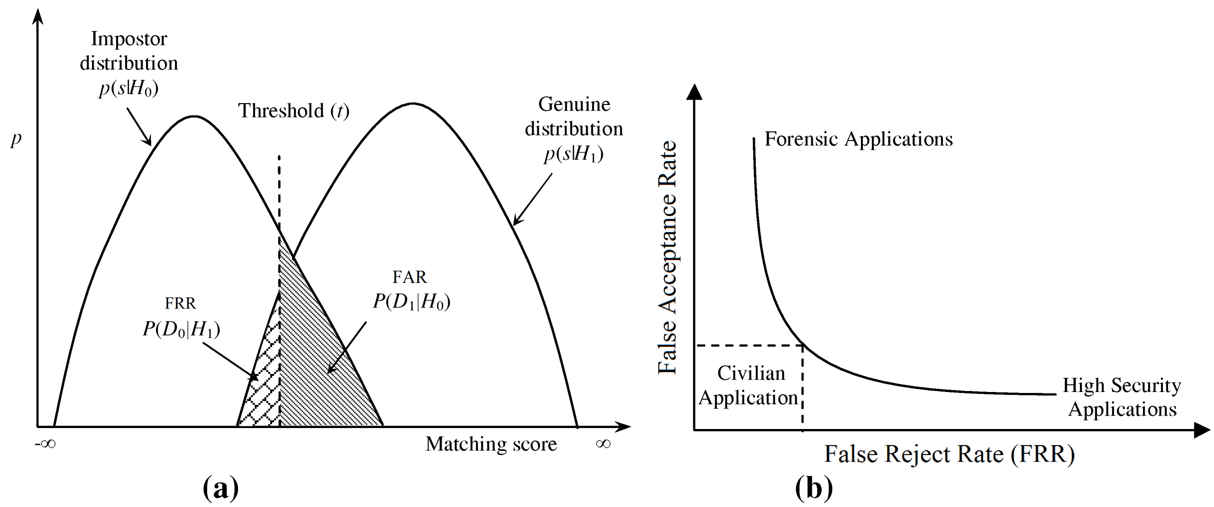


Figure 3.1: Biometric system error rates.(a) FAR and FRR for a given threshold t are displayed over the genuine and impostor score distributions; (b) The curve relating FAR to FRR at different thresholds is the receiver operating characteristics (ROC). Image from [31]

3.2 Representation of signatures

The representation for signatures remains very close to the representation of characters. It is explained in brief here again for the sake of continuity. For the training set of signatures, strokes are extracted by taking the points of velocity minima. For each of these thus extracted segments, the $5-D$ representation of $(r, x_0, y_0, \theta_s, \theta_e)$ is computed. This forms the basis from which the vocabulary for the Bag-of-Strokes representation is computed. This space is quantized using k-means clustering. We've found that, this representaion, for the application of signature verification, requires only 4-D representation *i.e.,* $(r, y_0, \theta_s, \theta_e)$ for satisfactory performance. This means that, given a fine enough quantization space, the knowledge of whether the stroke exists in the signature or not is important compared to knowing the exact x-location of it.

3.3 Distance measure

Support Vector Machines (SVM), as detailed in section A.2, has the distinct advantage of having good generalization performance in a high dimensional feature space. It does so by finding the linear hyperplane that maximizes the margin between the two classes. The proposed distance metric learning uses the parameters of SVM in learning the distance measure.

In general metric learning parlance, Mahalanobis distance is used to refer to a distance metric of the form

$$d_A(x, y) = (x - y)^T A(x - y),$$

where $A \succeq 0$ i.e., eigen-values of A are non-negative. The goal of metric-learning is to learn a *suitable* matrix A, such that distance between samples of the same class is as small as possible and distance between two samples of different classes is large. Various methods have been proposed to tackle the problem of metric learning i.e., computation of A. Comprehensive surveys on the subject can be found in [38, 73].

In the most basic interpretation, Mahalanobis distance can be understood as a distance between the two points in a linearly transformed space. The Eucleadean distance between two points x_1 & x_2 is given by $||x_1 - x_2||_2^2$. If L is a linear transformation that is applied to the space of $x_1 \& x_2$ then the distance between them can be computed as

$$d(Lx_1, Lx_2) = ||Lx_1 - Lx_2||_2^2 \tag{3.1}$$

$$= (Lx_1 - Lx_2)^T (Lx_1 - Lx_2) \tag{3.2}$$

$$= (x_1 - x_2)^T L^T L (x_1 - x_2) \tag{3.3}$$

$$= (x_1 - x_2)^T A (x_1 - x_2), \tag{3.4}$$

where $A = L^T L$.

Running the experiment in identification mode with an SVM trained on the representations proposed in 3.2, gives high accuracies ranging from $95.5\% - 98.7\%$ for varying number of clusters in the Bag-of-Strokes representation.

The output of classifier C_i of a trained SVM is of the form

$$C_i(x) = \mathbf{w}_i^T \mathbf{x} - b_i \ = v_i^T \tilde{x},$$

where $v_i = (w_i, -b_i)^T$ and $\tilde{x} = (\mathbf{x}\ 1)$. By concatenating all such $\binom{k}{2}$ vectors, we get the projection matrix V.

$$V = (v_1, v_2 \dots v_n)$$

This method is very efficient as solving for each v_i is independent of another and thus can be executed in parallel.

Using thus computed V as the projection vector in 3.4, we get the final distance as

$$d_g^2(x_i, x_j) = (\tilde{x}_i - \tilde{x}_j)^T V V^T (\tilde{x}_i - \tilde{x}_j) \tag{3.5}$$

The final metric matrix is computed as

$$M = V V^T$$

The equation 3.5 can be interpreted as follows. Let

$$g_i(\tilde{\mathbf{x}}) = V^T \tilde{\mathbf{x}}$$

The sign of $g_i(\tilde{\mathbf{x}})$ is the decision of the class of $\tilde{\mathbf{x}}$. Thus the distance between two samples is the correlation of the class labels of the two. Also, each of the n SVMs essentially localise the point in the representation space. Thus the transformation $V^T \times \tilde{x}$ can be understood has controlling the space that can be occupied by the point.

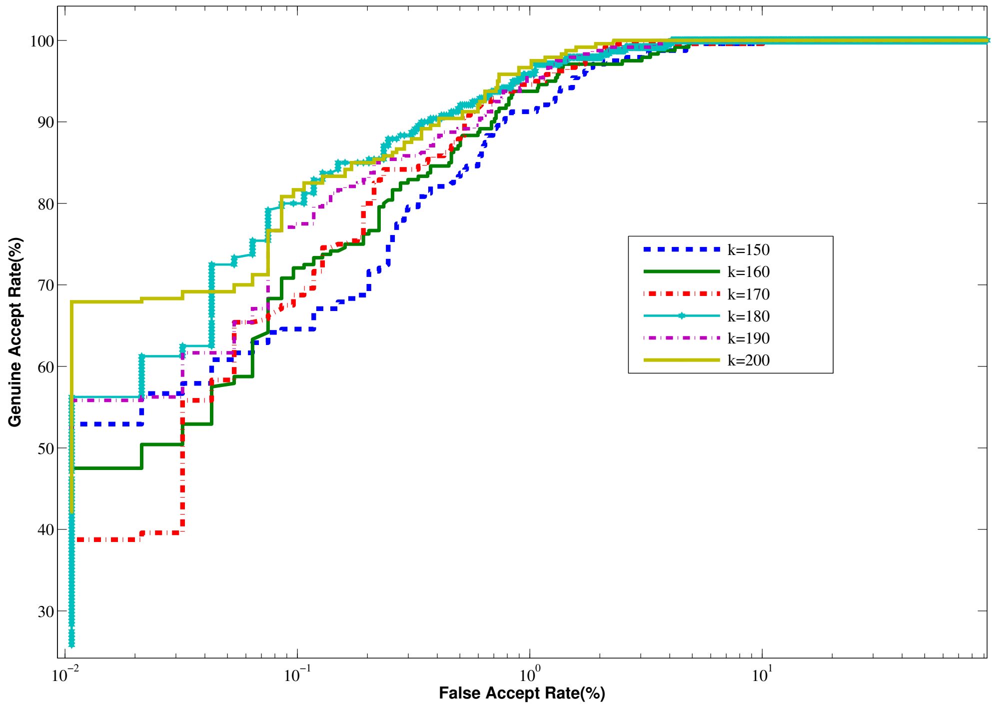


Figure 3.2: ROC (Receiver operating characteristic) curve for random forgeries for varying number of words in BoS representation. The EERs for each of the plots are 2.5%, 2.50%, 1.97%, 2.04%, 1.67%, 1.43% respectively. Plots are best viewed in colour on a monitor.

3.4 Results

3.4.1 Database

The publicly available database SVC-2004[74] was used for evaluation of our method. The dataset contains the signatures by 40 users each providing 20 repetitions of their signatures. The data was digitized with a WACOM Intuos tablet. The 20 signatures were collected in two sessions one week apart. Along with the 20 genuine signatures, 20 skilled forgeries were also collected from 4 contributors. The publicly available parts of the dataset have a total of 1600 signatures.

3.4.2 Experiments

Figures 3.2 & 3.3 show the *ROC* for varying number of clusters in the BoS representation. The similarity metric used to produce the plots shown is given by the Mahalanobis distance given by equation 3.5

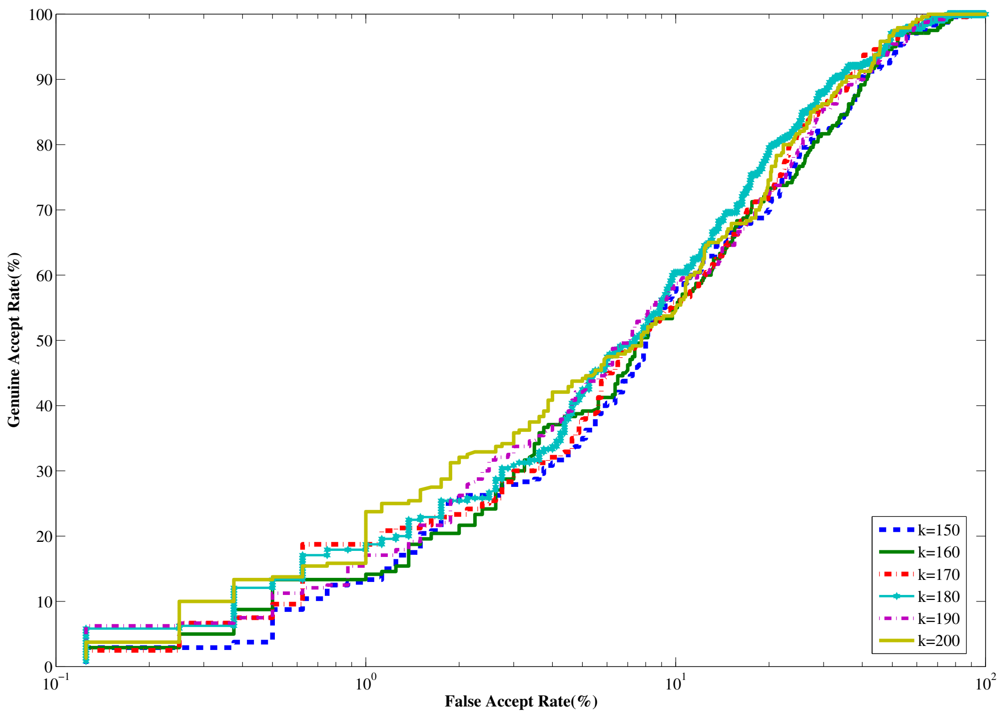


Figure 3.3: ROC (Receiver operating characteristic) curve for skilled forgeries for varying number of words in BoS representation. The EERs for each of the plots are 23.75%, 24.12%, 22.88%, 21.75%, 23.38%, 22.88% respectively. Plots are best viewed in colour on a monitor.

Data split	Equal Error	
	Random forgery	Skilled Forgery
10 training - 10 test	3.49%	27.25%
5 training - 15 test	3.98%	27.12%

Table 3.3: Changes in EERs with change in train-test splits

It has been shown in [36] that all the columns of the matrix V in equation3.5 *i.e.,* exhaustive calculation of the support vectors is not required to get satisfactory performance. This has been verified to see if that is the case with signatures too. The results are shown in figure 3.4, which show that in spite of discarding most of the support vectors ($(3/4)^{th}$), the performance doesn't degrade much.

The size of training set till here has been a split of 14-6 *i.e.,* 14 train samples and 6 test samples per class. To further elucidate the power of the representation and the metric learning algorithm, the number of training samples are changed and the resulting accuracies are shown in the table 3.3

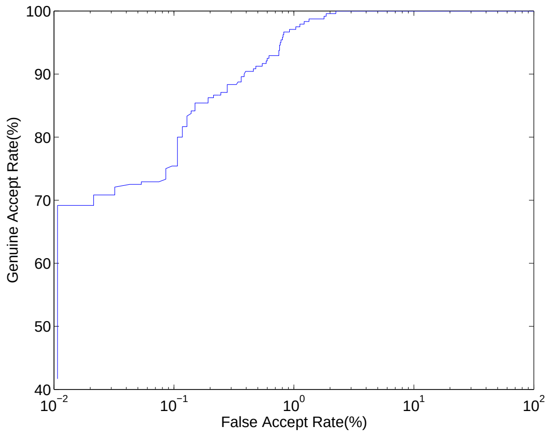
3.4.3 Comparison with other methods

The current method fares nearly equally when compared to several state-of-art algorithms as shown in table 3.4. We currently compare only random forgeries as the reason for this experiment is to prove that the current representation can robustly represent the spatial characteristics of signatures. Skilled forgeries are known to be more discriminable in velocity domain.

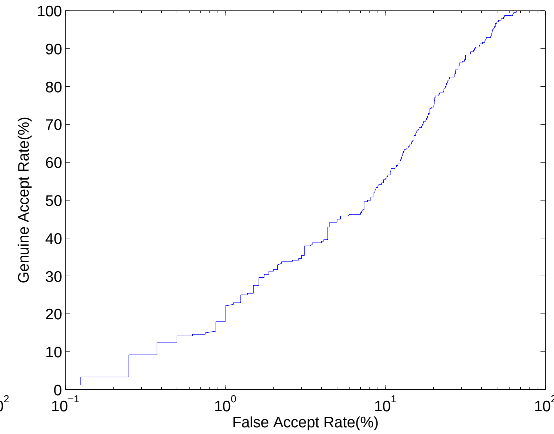
Fierrez *et al.* [19], uses two sub-systems and fuses the scores of the two sub-systems using simple fusion rules like minimum, maximum and product. For the classifier, standard features of velocity in x & y dimensions and pressure are used. Scoring function used is a Dynamic Time Warping (DTW) matching between database and query signatures which is normalized. The second system uses a Hidden Markov Model(HMM) based approach on additional feature sets like acceleration, curvature. Due to the use of multiple sub-systems, this system is able to perform better. It is important to note that, compared to other methods that use complex methods like HMMs, our proposed method is of fixed length representation and comparison is of much simpler computation.

Method	Error Rates
Yeung et al[74]	3.2%
Fierrez et al[18]	3.02%
Fierrez et al[19]	0.5%
Proposed method	1.4%

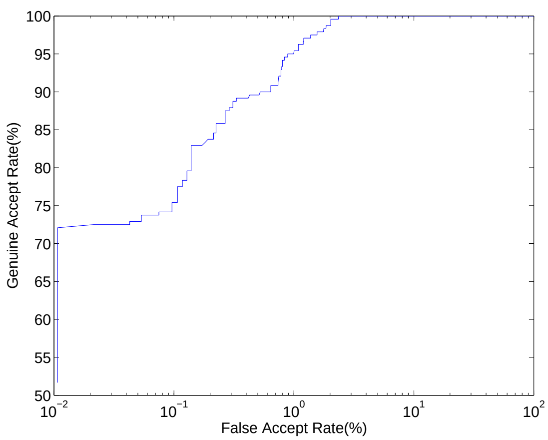
Table 3.4: EERs for random forgeries of various systems



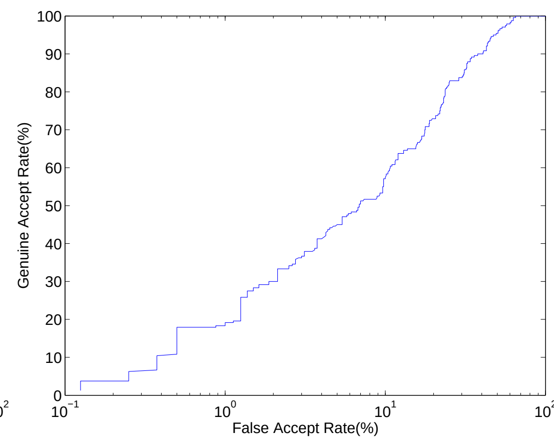
(a) 25% of the support vectors been removed - Random forgeries EER = 1.34%



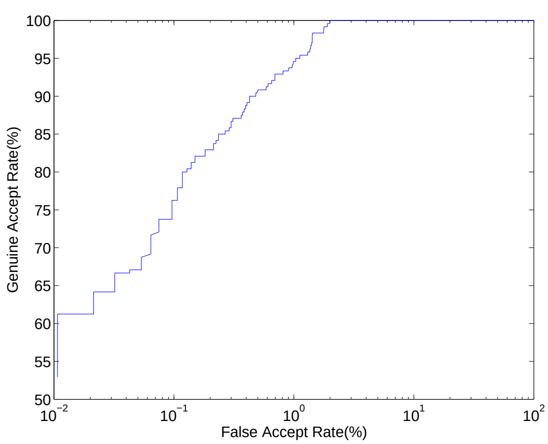
(b) 25% of the support vectors been removed - Skilled forgeries EER = 22.88%



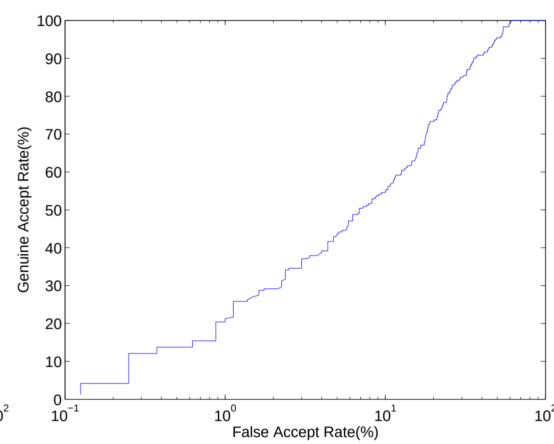
(c) 50% of the support vectors been removed - Random forgeries yields an EER of 1.77%



(d) 50% of the support vectors been removed - Skilled forgeries yields an EER of 22.88%



(e) 75% of the support vectors been removed - Random forgeries EER = 1.67%



(f) 75% of the support vectors been removed - Skilled forgeries EER = 22.75%

Figure 3.4: ROC curves for various number of classes used to construct the Metric matrix

3.4.4 User specific thresholds

Reiterating the previously stated fact that an identity claim is made and it is to be verified whether that claim is correct or not is the basic functional mode of a signature verification system. Thus is possible to have a user specific threshold as previously done in [19]. The table 3.5 illustrates the effect on EER, user specific thresholds has on the accuracy of the verification system.

Threshold type	**Equal Error Rate**	
	Random forgery	Skilled Forgery
User independent	1.43%	22.88%
User dependent	0.42%	17.5%

Table 3.5: EER for user-specific thresholds

3.5 Conclusion

In this chapter we extended the representation proposed for character recognition to the task of signature verification. We proposed a metric learning strategy that builds on pairwise classifier of a few randomly classes of the training set. We showed that this representation is very adept at discriminating random forgeries. We also showed that this representation is strong enough to discriminate skilled forgeries to an acceptable level.

Chapter 4

Conclusions

4.1 Summary

As the deluge of smart devices continues, it is essential that algorithms that deal with handwriting input on these systems are robust and fast. In this thesis, we proposed a method of representing handwriting in terms of its constituent ballistic strokes, which are the most fundamental units of handwriting. This representation is based on the very famous bag-of-Visual-Words model and has been termed as *bag-of-strokes*. We also proposed a curvature based segmentation method, as opposed to the traditional velocity minima based segmentation, and showed that this method of segmentation is more robust to noise. Recognition using this representation of characters is shown to exceed the state-of-art on multiple datasets.

We then extended this representation to the task of signature verification. Keeping the representation the same, we proposed a similarity metric based on metric learning. This metric learning algorithm learns from the separating hyperplanes of individual users of the biometric system. This strategy has the advantage of being scalable.

4.2 Future work

The work in this thesis is not the conclusion on the topic of handwriting representation, but only the first step in that direction. The following avenues can be explored based on the current ideas.

- In the current thesis, we worked only with discrete character sets for recognition. While the extension of this representation to continuous (cursive or semi-cursive) handwriting is not straight forward, it is a challenging problem to be investigated.

- Our choice of representation of ballistic strokes with arcs of a circle is based on neuro-muscular production characteristics. While this choice is theoretically justified, the possibility of using simpler functional forms for representation for recognition is to be investigated.

- Bag-of-words (BoW), while being a very powerful feature encoding strategy, has been surpassed in performance by various other methods in vision literature like Fisher vectors (FV), Vector of Locally Aggregated Descriptors (VLAD) for various large scale vision tasks. Their applicability to the tasks of handwriting is to be investigated.
- Signatures are most discriminative in the velocity space. The current method only considers the spatial trajectory of signatures. More robust systems can be built if velocity information can be incorporated into the representation.

Appendix A

Background

In this chapter, we present background theory that is common to the two online handwriting tasks, *i.e.,* handwriting recognition & signature verification, presented in this thesis. This will help in the understanding of various ideas presented in Chapters 2 and 3. Readers who are familiar with the basic ideas of machine learning like SVMs, clustering and Bag-of-words based representations can skip the rest of the chapter.

A.1 Machine learning paradigms

Machine learning, in general terms, is the study of computer systems that learn from experience [46]. Machine learning has been applied to various fields like Computer Vision, Natural language processing, Search engines, Medical diagnosis, Stock market analysis, DNA sequencing, Robotics, Affective computing, Recommender systems.

Every pattern recognition/machine learning task has a module that is common called the *feature extractor*. The original input *i.e.,* image, speech waveform, handwriting sample, has to be processed to transform it to a space of variables where the task of learning is easier to solve. Features are a stabler representation of the input data and learning is faster in the feature space. Feature extraction generally reduces the dimensionlity of the input data, thus has the interpretation of being a form of dimentionality reduction.

Machine learning algorithms can be organized into various catergories based on the outcomes or the learning methodology involved. Some of them are Supervised learning, Unsupervised learning, Semi-supervised learning, Reinforcement learning. We reserve only the first two for this discussion.

A.2 Support Vector Machines based Classification

Supervised learning or Classification is the task of learning a function from labelled data, where the labelled data is a pair of data vector and its correponding target value. For example, in the task of character recognition, this data is the input character image and the target value is the name of the character. In more formal terms, given a training set of $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and it corresponding

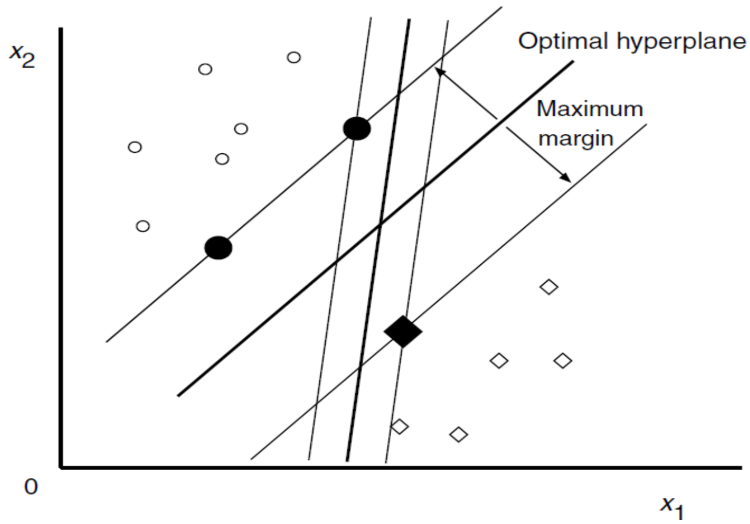


Figure A.1: Optimal max-margin hyperplane in two-dimensional space. Image from [2]

labels $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$, the goal of any classification algorithm is to find a mapping $f : \mathbf{X} \rightarrow \mathbf{Y}$ Various algorithms have been proposed in literature like Naive Bayes, Neural Networks, Decision trees, Random Forests. Support vector machines[2] is also type of supervised algorithm, which will be elucidated next.

Given a training set of L training samples

$$\{\mathbf{x_i}, y_i\}_{i=1}^{L},$$

where $y_i \in \{-1, +1\}$ and $x \in \mathcal{R}^d$. This y_i implies that the current classification task is that of a binary classification. The extension of this to a multi-class problem is dealt later. If the data is linearly separable, a line of the form $\mathbf{w}.\mathbf{x} + b = 0$ can separate the classes. This can be represented as

$$y_i(\mathbf{w}.\mathbf{x_i} + b) > 0 \, \forall i$$

If the data is linearly separable, it can be proved that infinitely many hyperplanes exist that separate the two classes. The best hyperplane of these is defined as the one that maximises the margin of separation as shown in figure A.1.

If a minimum separation margin of ρ has to be ensured, then that equation becomes

$$y_i(\mathbf{w}.\mathbf{x_i} + b) > \rho$$

This can be rewritten as

$$y_i(\mathbf{w}.\mathbf{x_i} + b) \geq 1 \tag{A.1}$$

with both $\mathbf{w}, b$ appropriately scaled.

Thus the margin of separation between the two classes can be shown to be $\frac{2}{|\mathbf{w}|^2}$. This separation has been to be maximized, along with the correct classification of all the training samples. Both of them can be put together into a single equation as shown in

$$\min \frac{||\mathbf{w}||^2}{2} \quad s.t \;\; y_i(\mathbf{w}.\mathbf{x_i}+b) > 1 \;\; \forall i \tag{A.2}$$

This can be solved with the help of Lagrangian multipliers. The Lagrangian primal is computed as

$$L_P \equiv \frac{1}{2}||w||^2 - \sum_{i=1}^{L} \alpha_i[y_i(\mathbf{x_i}.\mathbf{w}+b)-1] \tag{A.3}$$

$$\equiv \frac{1}{2}||w||^2 - \sum_{i=1}^{L} \alpha_i[y_i(\mathbf{x_i}.\mathbf{w}+b)] + \sum_{i=1}^{L}\alpha_i \tag{A.4}$$

The condition is that the Lagrangian multipliers $\alpha_i \geq 0$

Differentiating L_P with w, b

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x_i} \tag{A.5}$$

$$\frac{\partial L_p}{\partial b} = 0 \implies \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{A.6}$$

By substituting A.6 into A.4, the Lagrangian dual is computed to be

$$L_D \equiv \max_{\alpha} \left[\sum_{i=1}^{L} \alpha_i\right] - \frac{1}{2}\alpha^{\mathbf{T}}\mathbf{H}\alpha \;\; \alpha_i \geq 0 \;\; \forall\, i \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{A.7}$$

where $\mathbf{H} \equiv y_i y_j \mathbf{x_i}.\mathbf{x_j}$.

Support vectors are those $\mathbf{x_i}$ for which equation A.1 holds the equality *i.e.,* $y_s(\mathbf{x_s}.\mathbf{w}+b) = 1$. It is evident from equation A.6 that the normal vector to the separating hyperplane, also referred to as the weight vector $\mathbf{w}$ can be expressed as a the sum of subset of input vectors. This is a convex quadratic optimization problem, and a Quadratic Programming (QP) solver will return α and hence w.

SVMs have the implicit assumption that the data is linearly separable. This can be relaxed by the use of slack-variables at the margin of classification, which is the equivalent of stating of tolerance ζ_i. The objective function is written as

$$\min \frac{||\mathbf{w}||^2}{2} + C\sum_{i=L}^{M} \zeta_i \quad s.t \;\; y_i(\mathbf{w}.\mathbf{x_i}+b) > 1-\zeta_i \;\; \forall i \tag{A.8}$$

Lagrangian looks very similar to the A.7 except that the α_i's have an upper limit on the regularization parameter C.

Extension to multiple-class classification problem is possible in two ways.

- One vs one: A k-class problem is broken down into $\binom{k}{2}$ two class problems by considering all possible combinations and the final class is decided based on majority voting.
- One vs rest: A k-class problem is broken down into k problems such that, one class is discriminated against rest of the $k-1$ class problems. This results in k binary classifiers.

In our experiments we consider the one vs one formulation.

Application of SVMs

In order to apply SVMs the vector $\mathbf{w}$ and b have to be computed. The steps for linearly separable case of SVMs with and without slack variables is very minor and is mentioned in point 2 below.

- Compute the matrix $\mathbf{H}$, where $H_{ij} = y_i y_j \mathbf{x_i^T x_j}$

- Find α, such that

$$\left[\sum_{i=1}^{L} \alpha_i\right] - \frac{1}{2}\alpha^{\mathbf{T}}\mathbf{H}\alpha$$

  is maximized, with the constraints

$$\alpha_i \geq 0 \,\forall i \,\&\, \sum_{i=1}^{L} \alpha_i y_i = 0$$

  This can be done, as mentioned previously, by a QP solver. For the case with slack variables, the condition $\alpha_i \geq 0$ changes to $0 \leq \alpha_i \leq C$ where C is the weight given to the slack variables in the formulation.

- Calculate

$$\mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x_i}$$

- Determine the set of support vectors $\mathbb{S}$, which are the set of input training vectors for which $\alpha_i \geq 0$

- Calculate

$$b = \frac{1}{N_s} \sum_{s \in \mathbb{S}} \left(y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x_m}.\mathbf{x_s} \right)$$

  With this computation, as both parameters of the hyperplane $\mathbf{w}, b$ are computed.

- When a new test sample comes, it is classified using

$$y' = sgn(\mathbf{w}.\mathbf{x}' + b)$$

  where sgn is the signum function defined as

$$sgn(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

Implications of large margin

- One of the major advantages of a traditional SVM formulation is that owing to the quadratic program the error surface is convex and thus has a global optimum.

- Outliers are a tough problem to deal with, in a machine learning problem. The margin parameter C controls the misclassification error. If C is set to a large value, misclassification is suppressed, and if it is set to a small value, training data that are away from the gathered data are allowed to be misclassified. With a proper choice of the parameter C, the effects of outliers can be reduced to a reasonable extent.

- Expected risk of misclassification is generally written as a sum of empirical risk (training set error) and a function of VC (Vapnik-Chervonenkis) dimension. In simplistic terms, VC dimension is a measure of free parameters of the model. In the case of SVMs, it has been shown that maximizing the margin results in the decrease of VC dimension. Since the separating hyperplane has zero empirical error (it correctly separates all the training examples), maximizing the margin will also minimize the upper bound on the expected risk. Thus there are theoretical guarantees of the functioning of SVMs

A.3 K-means Clustering

In the previous section A.2, the data was presented along with its label. It is also possible to pose the problem of finding patterns in data without the labels, *i.e.,* finding groupings in data such that data in the same group are similar to each other. This pattern finding is dependent on the definition of similarity. In this section, we examine a famous model of unsupervised clustering called k-means clustering. K-means group a given set of n points by assigning each data point to one of the k clusters based on minimum distance.

Given n data points $x_1, x_2, \dots x_n$ where $x_i \in \mathcal{R}^D$, and k number of clusters ($k < n$).
$\mathbf{C} = \mathbf{C_1}, \mathbf{C_2}, \dots \mathbf{C_k}$ be the k clusters that that contain some of the data points. μ_i be the mean of the i^{th} cluster, given by

$$\mu_i = \frac{\sum_{x_j \in C_i} x_j}{|C_i|}$$

The objective function of k-means, which it tries to minimize is

$$C = arg \min_C \sum_{i=1}^{k} \sum_{x_j \in C_i} ||x_j - \mu_i||^2 \tag{A.9}$$

The minimization of A.9, is done iteratively over two steps and starts by randomly initializing the k means $\mu_1, \mu_2, \dots \mu_k$.

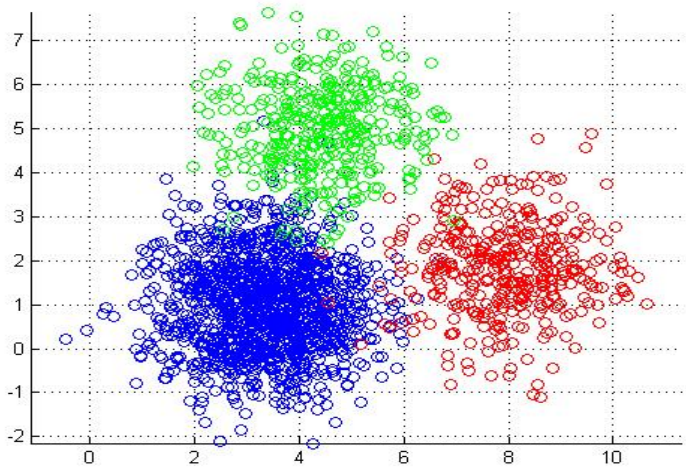


Figure A.2: Without being given the class labels of the data, it is obvious that there are 3 clusters in the data. This is the basis of k-means algorithm

- **Step 1:** Each data point is assigned to the cluster whose mean is closest to that point.

$$C_i = \{x_p : ||x_p - \mu_i|| \leq ||x_p - \mu_j||\} \forall j$$

- **Step 2:** The mean of each cluster is recomputed according to the cluster assignments in the 'E' step.

$$\mu_i = \frac{\sum_{x_j \in C_i} x_j}{|C_i|}$$

The algorithm is said to have converged when the assignments do not change. Sometimes, the algorithm is also stopped based on a threshold on the number of iterations or on a tolerance on the objective function in equation A.9. There are no theoretical guarantees of the convergence of the algorithm. K-means is generally very sensitive to the initial centroids provided. The only parameter to be set if the number of clusters k, which is generally set through cross-validation.

A.4 Bag of words model

The bag of words was originally proposed in the domain of textprocessing[61] where a document is represented an unordered collection of words. The order in which the words occured in an sentence has no consequence on the feature representation. For example, the phrases "task of english handwriting recognition" and "english task handwriting of recognition" will have the same feature representation. The first step of the process is choosing the "words". In the above example, if the words chosen are *english, recognition, task, cricket, minister, cinema*, the word histogram is formed as $(1, 1, 1, 0, 0, 0)$

i.e., there is one instance of the word *english*, one of *recognition*, one of *task*, none of *cricket, minister,cinema*.

This idea has been extended and applied to various domains including computer vision. One of the first applications of BoW technique to vision problems is in [14] and is commonly referred to as Bag-of-Visual-Words model. The first step in it is building the equivalent of words in visual domain. That is done by taking a large set of object images and extracting descriptors (like SIFT or SURF) from detected keypoints and clustering them using k-means. The cluster centers are considered visual words. This is illustrated in figureA.3[1]

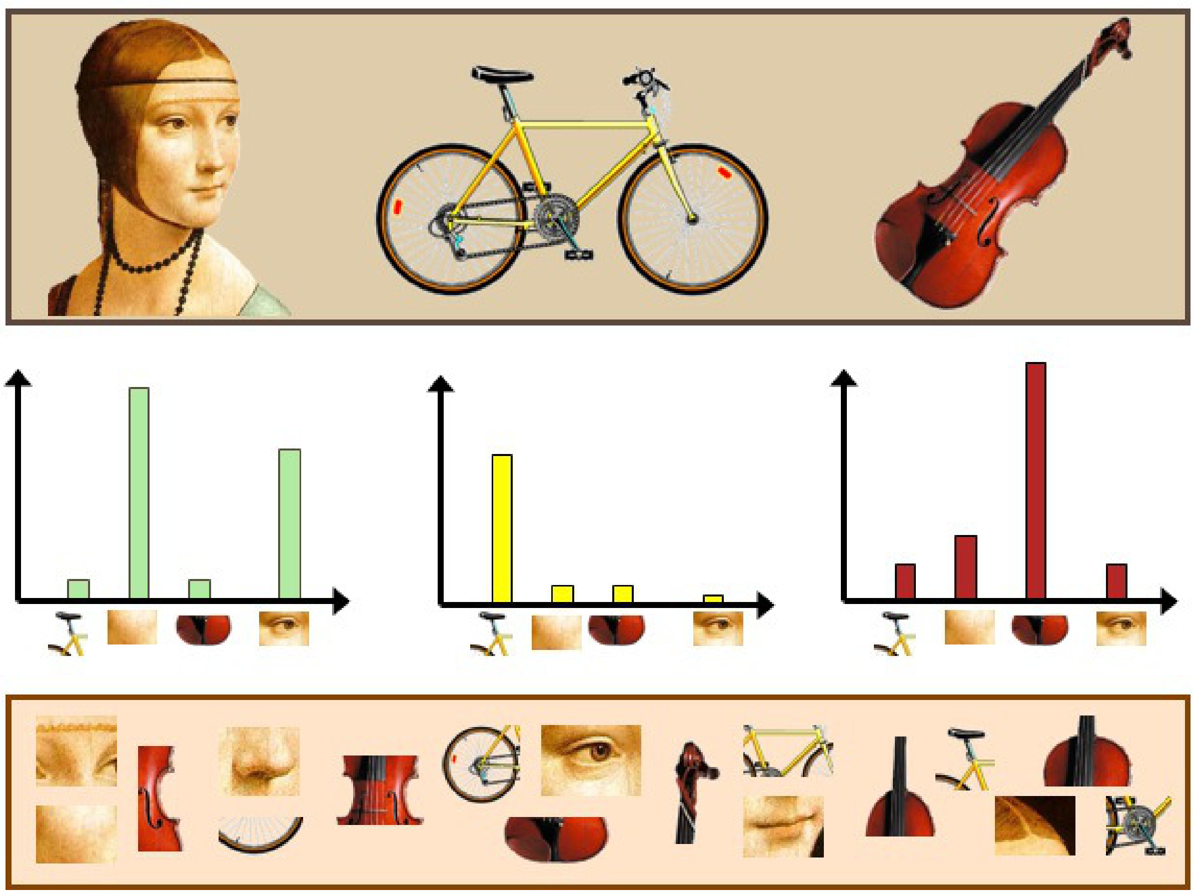


Figure A.3: Illustration of training of Bag-of-Words(BoW) model

Given a new image, features are extracted as in the training step. For each descriptor extracted its nearest neighbor in the dictionary is computed. A histogram is built of length k(number of words) where the j^{th} value is the frequency of the j^{th} dictionary word.

These histograms from the training set are used to train a classifier, usually an SVM or Naive Bayes and the during the test time as with the example shown in A.4, representation is built and tested. This

[1]Images from gilscvblog.wordpress.com/2013/08/23/bag-of-words-models-for-visual-categorization/

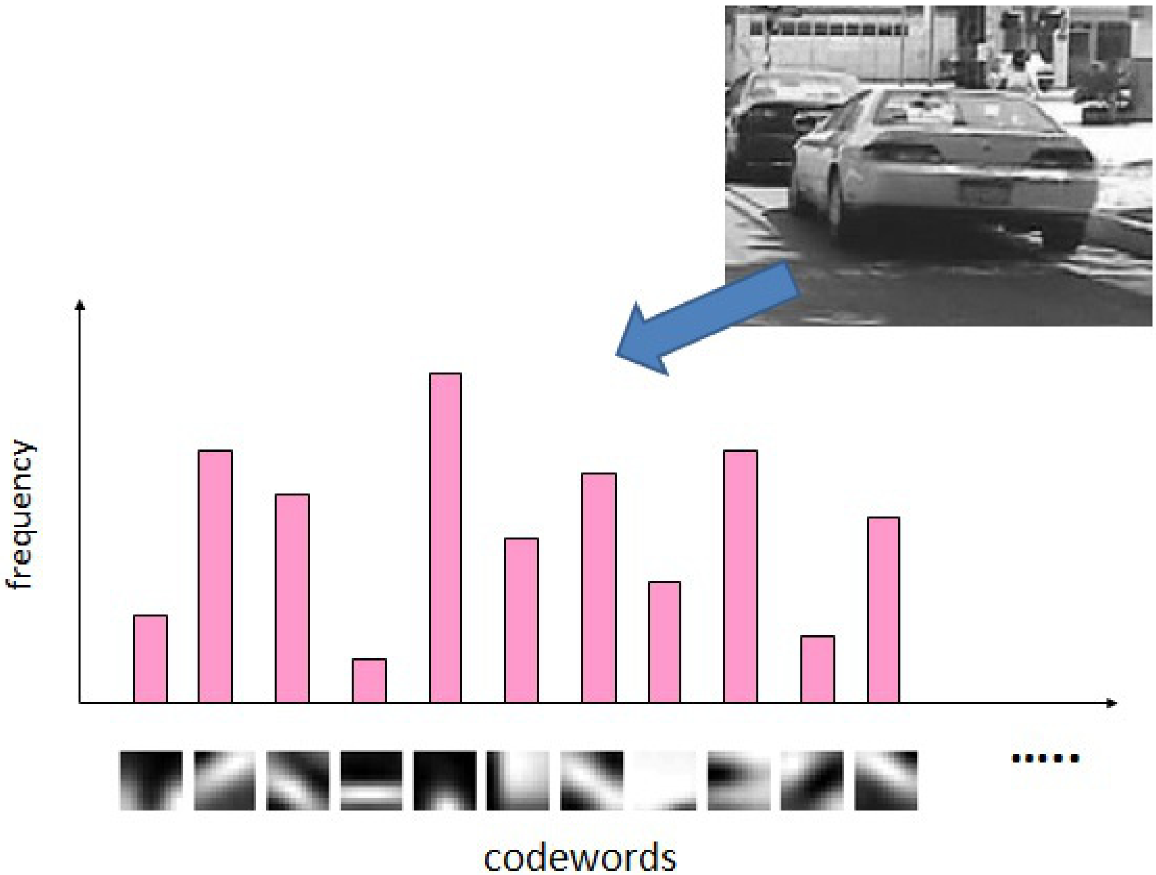


Figure A.4: Illustration of testing with Bag-of-Words(BoW) model

method has been applied to great success in object recognition, scene categorization, image retrieval [69]

To the best of our knowledge, the BoVW model has not been used in the context of online handwriting.

Related Publications

Prabhu Teja .S, and Anoop M. Namboodiri. "A Ballistic Stroke Representation of Online Handwriting for Recognition." Document Analysis and Recognition (ICDAR), 2013 12^{th} International Conference on. IEEE, 2013.

Bibliography

[1] Google Nexus 7 tablet. `http://www.google.co.in/nexus/7/specs/`. Accessed: 26-Dec-2013.

[2] S. Abe. *Support vector machines for pattern classification*. Springer, 2010.

[3] A. Arora and A. M. Namboodiri. A hybrid model for recognition of online handwriting in indian scripts. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 433–438. IEEE, 2010.

[4] C. K. Au-Yeung and D. J. Love. On the performance of random vector quantization limited feedback beamforming in a miso system. *IEEE Transactions on Wireless Communications,*, 6(2):458–462, 2007.

[5] N. Balakrishnan, R. Reddy, M. Ganapathiraju, and V. Ambati. Digital library of india: a testbed for indian language research. *TCDL Bulletin*, 3(1):No–pp, 2006.

[6] L. K. Baxter. *Capacitive Sensors: Design and Applications*. IEEE Press, Piscataway, NJ, 1996.

[7] E. J. Bellegarda, J. R. Bellegarda, D. Nahamoo, and K. S. Nathan. A discrete parameter hmm approach to on-line handwriting recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2631–2634. IEEE, 1995.

[8] H. Bezine, A. M. Alimi, and N. Sherkat. Generation and analysis of handwriting script with the beta-elliptic model. In *Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 515–520. IEEE, 2004.

[9] R. Bolle. *Guide to biometrics*. Springer, 2004.

[10] H. Bunke and P. Wang. *Handbook Of Character Recognition And Document Image Analysis*. World Scientific, Hackensack, NJ, 1997.

[11] M. Castro-Bleda et al. Improving a DTW-based recognition engine for on-line handwritten characters by using MLPs. In *10th International Conference on Document Analysis and Recognition*, Barcelona, Spain, July 2009.

[12] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[13] R. Clarke. Human identification in information systems: Management challenges and public policy issues. *Information Technology & People*, 7(4):6–37, 1994.

[14] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2, 2004.

[15] C. De Boor. A practical guide to splines. 2001.

[16] M. Djioua and R. Plamondon. A new algorithm and system for the characterization of handwriting strokes with delta-lognormal parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 31(11):2060–2072, 2009.

[17] J. Fierrez and J. Ortega-Garcia. *On-line signature verification*, pages 189–209. Springer, 2008.

[18] J. Fierrez, J. Ortega-Garcia, D. Ramos, and J. Gonzalez-Rodriguez. Hmm-based on-line signature verification: Feature extraction and signature modeling. *Pattern recognition letters*, 28(16):2325–2334, 2007.

[19] J. Fierrez-Aguilar, S. Krawczyk, J. Ortega-Garcia, and A. K. Jain. Fusion of local and regional approaches for on-line signature verification. In *Advances in Biometric Person Authentication*, pages 188–196. Springer, 2005.

[20] G. Gangadhar, D. Joseph, and V. Chakravarthy. An oscillatory neuromotor model of handwriting generation. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(2):69–84, 2007.

[21] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernández. Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 577–584, 2007.

[22] I. Guyon, J. S. Denker, and Y. LeCun. Time delay neural network for printed and cursive handwritten character recognition, Apr. 14 1992. US Patent 5,105,468.

[23] I. Guyon, D. Henderson, P. Albrecht, Y. LeCun, and J. Denker. Writer independent and writer adaptive neural network for on-line character recognition. *From Pixels to Feature III: Frontiers in Handwriting Recognition. Elsevier Science Publishers*, 1992.

[24] M. Hart. *Project gutenberg*. Project Gutenberg, 1971.

[25] J. Hollerbach. An oscillation theory of handwriting. *Biological Cybernetics*, 39(2), Mar. 1981.

[26] J. Hu, M. K. Brown, and W. Turin. HMM based online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.

[27] D. Impedovo and G. Pirlo. Automatic signature verification: the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(5):609–635, 2008.

[28] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online handwriting recognition: the npen++ recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–180, 2001.

[29] A. K. Jain, F. D. Griess, and S. D. Connell. On-line signature verification. *Pattern recognition*, 35(12):2963–2972, 2002.

[30] A. K. Jain, A. Ross, and K. Nandakumar. *Introduction to biometrics*. Springer, 2011.

[31] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology,*, 14(1):4–20, 2004.

[32] R. S. Kashi, J. Hu, W. Nelson, and W. Turin. On-line handwritten signature verification using hidden markov model features. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition,*, volume 1, pages 253–257. IEEE, 1997.

[33] R. H. Kassel. *A comparison of approaches to on-line handwritten character recognition*. PhD thesis, Massachusetts Institute of Technology, 1995.

[34] M. Kherallah, F. Bouri, and A. M. Alimi. On-line arabic handwriting recognition system based on visual encoding and genetic algorithm. *Engineering Applications of Artificial Intelligence*, 22(1), Feb. 2009.

[35] M. Kherallah, L. Haddad, A. M. Alimi, and A. Mitiche. On-line handwritten digit recognition based on trajectory and velocity modeling. *Pattern Recognition Letters*, 29(5), Apr. 2008.

[36] T. Kozakaya, S. Ito, and S. Kubota. Random ensemble metrics for object recognition. In *IEEE International Conference on Computer Vision (ICCV),*, pages 1959–1966. IEEE, 2011.

[37] E. Kreyszig. *Differential Geometry*, chapter Principal Normal, Curvature, Osculating Circle. Dover Books, 1991.

[38] B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013.

[39] F. Lacquaniti, C. Terzuolo, and P. Viviani. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54(1-3), Oct. 1983.

[40] F. Leclerc and R. Plamondon. Automatic signature verification: the state of the art1989–1993. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(03):643–660, 1994.

[41] H. Lei and V. Govindaraju. A comparative study on the consistency of features in on-line signature verification. *Pattern Recognition Letters*, 26(15):2483–2489, 2005.

[42] C.-Y. Liou and H.-C. Yang. Handprinted character recognition based on spatial topology distance measurement. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 18(9):941–945, 1996.

[43] D. Llorens et al. The UJIpenchars database: A pen-based database of isolated handwritten characters. In *Proceedings of the Sixth ILRE*, Marrakech, Morocco, May 2008.

[44] A. Mauceri. Feasibility study of personnel identification by signature verification. Technical report, DTIC Document, 1965.

[45] B. Miller. Vital signs of identity [biometrics]. *Spectrum, IEEE*, 31(2):22–30, 1994.

[46] T. M. Mitchell. Machine learning. wcb, 1997.

[47] A. M. Namboodiri. *On-line Handwritten Document Understanding*. PhD thesis, Michigan State University, East Lansing, MI, 2004.

[48] A. M. Namboodiri and A. K. Jain. Robust segmentation of unconstrained online handwritten documents. ICVGIP, 2004.

[49] NDL. Library of congress: National digital library., Jan. 2014.

[50] C. O'Reilly and R. Plamondon. Development of a sigmalognormal representation for on-line signatures. *Pattern Recognition*, 42(12):3324–3337, Dec. 2009.

[51] M. Parizeau and R. Plamondon. A fuzzy-syntactic approach to allograph modeling for cursive script recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 17(7):702–712, 1995.

[52] R. Plamondon. A kinematic theory of rapid human movements. *Biological Cybernetics*, 72(4):295–307, 1995.

[53] R. Plamondon. A kinematic theory of rapid human movements: Part III. kinetic outcomes. *Biological Cybernetics*, 78(2):133–145, 1998.

[54] R. Plamondon, C. Feng, and A. Woch. A kinematic theory of rapid human movement. part IV: a formal mathematical proof and new insights. *Biological Cybernetics*, 89(2):126–138, 2003.

[55] R. Plamondon and W. Guerfali. The generation of handwriting with delta-lognormal synergies. *Biological Cybernetics*, 78(2):119–132, 1998.

[56] R. Plamondon and G. Lorette. Automatic signature verification and writer identificationthe state of the art. *Pattern recognition*, 22(2):107–131, 1989.

[57] R. Plamondon and F. J. Maarse. An evaluation of motor models of handwriting. *IEEE Transactions on Systems, Man and Cybernetics,*, 19(5):1060–1072, 1989.

[58] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 22(1):63–84, 2000.

[59] J. C. Platt and N. P. Matic. A constructive rbf network for writer adaptation. In *Advances in neural information processing systems*. Citeseer, 1997.

[60] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[61] G. Salton and M. J. McGill. Introduction to modern information retrieval. 1983.

[62] W. Santipach and M. L. Honig. Asymptotic performance of mimo wireless channels with limited feedback. In *Military Communications Conference,*, volume 1, pages 141–146. IEEE, 2003.

[63] P. Scattolin. *Recognition of handwritten numerals using elastic matching*. PhD thesis, Citeseer, 1995.

[64] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time-delay neural networks and hidden markov models. *Machine Vision and Applications*, 8(4):215–223, 1995.

[65] L. Schomaker. From handwriting analysis to pen-computer applications. *Electronics & Communication Engineering Journal*, 10(3):93–102, 1998.

[66] A. J. Sellen and R. Harper. *The myth of the paperless office*. The MIT Press, 2003.

[67] T. Shibata, Y. Tonouchi, S. Kubota, T. Nakai, and Y. Yamaji. Fast and memory efficient online handwritten strokes retrieval using binary descriptor. In *IAPR Asian Conference on Pattern Recognition (ACPR),*, pages 647–651. IAPR, 2013.

[68] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition methods. In *IEEE International Conference on Acoustics, Speech, and Signal Processing,*, pages V–125. IEEE, 1994.

[69] R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[70] C. C. Tappert, C. Y. Suen, and T. Wakahara. Online handwriting recognition-a survey. In *International Conference on Pattern Recognition,*, pages 1123–1132. IEEE, 1988.

[71] C. C. Tappert, C. Y. Suen, and T. Wakahara. The state of the art in online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 12(8):787–808, 1990.

[72] G. P. Van Galen and J. F. Weber. On-line size control in handwriting demonstrates the continuous nature of motor programs. *Acta Psychologica*, 100(1):195–216, 1998.

[73] L. Yang. Distance metric learning: A comprehensive survey. 2006.

[74] D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll. SVC2004: First international signature verification competition. In *Biometric Authentication*, pages 16–22. Springer, 2004.