Chapter 2

Literature Review

This chapter presents the state of the art in handwriting recognition technology with an emphasis on Online Handwriting Recognition of Assamese characters. Online and Offline Handwriting Recognition is discussed. We discuss issues related to datasets, online text input devices and online data acquisition methodology. We present an overview of preprocessing, feature computation and classification techniques, which are the three main stages of any character recognition system. This chapter further highlights a few datasets of online handwritten characters for Western Scripts as well as Indian Scripts.

2.1 Handwriting recognition

Handwriting recognition is the process of transforming a language which is represented in its spatial form of graphical marks into its symbolic representation [2]. Handwriting properties and issues related to problems involved in the machine recognition of handwriting are explained by Tappart et al. [1]. Plamondon et al. [2] describe the nature of handwritten language and how it is transduced into electronic data and highlight application areas of both online and offline recognition systems. Applications of Online Handwriting Recognition (OHR) are found in handheld computers such as PDAs and Tablet PCs. The success of online systems opens the idea of developing offline systems that first estimate the trajectory of the writing from offline data and then use online recognition algorithm [2].

2.1.1 Online and Offline Handwriting Recognition

Two different approaches to handwriting recognition are online and offline recognition. Online recognition refers to the methods and techniques dealing with the automatic processing of a character as it is written using a digitizer [2]. In online handwriting recognition system, recognition is performed at the moment the writer writes. OHR involves automatic conversion of text as it is written on a special digitizer, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. The obtained signal is converted into letter codes which are usable within computer and text-processing applications. The elements of an OHR interface typically includes a pen or stylus for the user to write with, a touch sensitive surface, which may be integrated with, or adjacent to, an output display and a software application which interprets the movements of the stylus across the writing surface translating the resulting curves into digital text. The online data contains the temporal information about the writing process, in addition to the spatial shape information of the characters. A recognition process can make use of both the above information for robust performance. On the other hand, in offline system, recognition is performed after the digitized image of the handwritten document is captured by a scanner. Offline Handwriting Recognition system is a subfield of Optical Character Recognition (OCR), where OCR is the mechanical or electronic conversion of scanned or photographed images of printed or handwritten text into computer readable text. Online recognition is more accurate than that of offline due to the additional temporal information [1].

2.1.2 Writer Dependent and Writer Independent Handwriting Recognition

Handwriting recognition systems can be classified into writer dependent and writer independent recognition systems. Writer dependent systems are developed for recognizing a specific or known pattern of handwriting. Writer dependent recognition systems are trained and tested for a specific group of writers and the system aims at recognizing the handwritings of only those writers for which it is trained and tested. Writer dependent recognition systems deal with lower variability in handwriting, which leads to higher recognition rate in such systems. On the other hand, writer independent recognition systems are developed for recognizing handwritings with unknown patterns. In this case, the recognition system has to deal with high variability in patterns of handwriting, since the system is not limited to a specific group of writers. In the category of writer independent recognition systems, recognition is performed on the handwritings of those writers for which the system is not trained. Writer independent recognition systems are difficult to develop in comparison to writer dependent recognition systems.

2.1.3 Constrained and Unconstrained Handwriting Recognition

Handwriting recognition systems can also be classified as constrained and unconstrained systems. Constrained handwriting is concerned with where and how characters are written. In constrained handwriting recognition system, writing of each character is constrained to a specific number, order and direction of strokes. Characters may be written in predefined boxes to control both vertical and horizontal spacing. There is flexibility of writing in unconstrained handwriting [1]. Writing of each character can be highly variable in unconstrained writing for each individual writer as there are no such specific instructions in writing as described above for constrained handwriting. Writings tend to be cursive in unconstrained system. Due to writing variations, handwriting recognition is difficult in case of unconstrained recognition systems.

2.1.4 Variability in Writing Styles

Writing style is mainly person dependent. Apart from that, factors that influence writing styles are writing surface of the tablet, writing posture, writing environment etc. Each individual character is written by different writers in different ways. Even the same writer tends to write a character differently at different times. Characters can vary in terms of the position at which these are written in the writing surface. Moreover, the characters can have variability in terms of size and direction of strokes.

2.2 Data Acquisition and Datasets

This section presents a brief discussion on data acquisition for the purpose of online handwriting recognition experiments and highlights a few standard datasets available in Western and Indian Scripts. Development of any recognition system requires a standard dataset which is used for training and testing of the system. Acquisition and distribution of standard datasets have increasingly gained importance.

2.2.1 Data Acquisition

The typical format of online handwriting data is a sequence of (X, Y) coordinate points (horizontal and vertical coordinates). Online handwriting data is captured by writing with a special pen called stylus on an electronic surface such as a digitizer combined with a liquid crystal display. The data in online handwriting is spatio-temporal in nature [2]. The two dimensional coordinates of successive points are stored as a function of time. Apart from the capture of these successive coordinate points, the data acquisition program typically captures two actions of the electronic pen as the writer writes on the digitizer surface. These two pen actions are Pen-down and Pen-up. Pen-down action is captured when the pen touches the digitizer surface at the start of writing and the pen-up action is captured when the pen is lifted at end of writing. A stroke is a sequence of (X, Y) coordinate points captured between a pen-down and a pen-up action. In case of single stroke characters we have occurrences of pen-down and pen-up actions only for once. In case of characters composed of multiple strokes we have the occurrences of pen-down and pen-up actions for multiple times which is equal to the number of strokes of the characters. While writing, writer applies pressure with the pen tip on the surface of the digitizer. The acquisition program may also capture the pressure of the pen tip at each coordinate point which the writer applies while writing on the pressure sensitive surface of the digitizer.

2.2.1.1 Text Input Devices

The common choices of online handwriting input devices are Personal Digital Assistants (PDAs), Pen Tablets and Tablet PCs. PDA, also known as a palmtop computer is a mobile device that functions as a personal information manager. A typical PDA has a touch screen for entering text with a stylus. A virtual keyboard is the typical method of entering text on touch screen, where a keyboard is shown on the touch screen. Text can also be entered by tapping the on-screen keyboard with a stylus. PDAs are equipped with the facility of handwriting recognition, where words or letters are written on the touch screen. Subsequently, the PDA converts the input to text. Stroke recognition facility allows the user to make a predefined set of strokes on the touch screen, sometimes in a special input area. This predefined set of strokes represents various characters to be input. The strokes are often character shapes in simplified form which mak them easier for the device to recognize. Palm's Graffiti is one such widely known stroke recognition system. A typical PDA is shown in Figure. 2.1.

Figure 2.1. Personal Digital Assistant

A pen tablet (or graphics tablet) or digitizing tablet is a computer that enables the user to draw images and graphics. It is similar to the way a person draws images with a pencil and paper. These tablets can be used to capture handwritings. The device consists of a flat surface on which the user can trace or draw an image using an attached stylus. The image is displayed on the computer monitor. Some graphics tablets also have a screen. Some tablets are used as a replacement for the mouse in desktop computers. A typical pen tablet is shown in Figure 2.2.

Figure 2.2. Wireless Pen Tablet (or Digitizing Tablet)

A tablet PC is a mobile computer. It has a display, circuitry and battery in a single unit. Tablets are equipped with a touch sensitive screen. Stylus or finger gestures can be used as substitutes for mouse and keyboard. Tablets also include physical buttons and ports. They usually have on-screen, pop-up virtual keyboards for typing. Tablets are normally larger than personal digital assistants. Tablets can be classified into several categories based on the presence and physical appearance of keyboards. Booklets and slates are without a physical keyboard and text input is done through the use of a virtual keyboard projected on a touch screen-enabled display. Physical keyboards are present in hybrids and convertibles. But virtual keyboards are also available in these devices. A typical tablet PC is shown in Figure 2.3.

Figure 2.3. Tablet PC

2.2.1.2 Data Acquisition Methodology

In the context of OHR, data acquisition methodology describes the systematic steps of collection of online handwriting samples. The first step in data acquisition methodology is the selection of online handwritten characters to be collected. A list of attributes or requisite information (for example, pen-down and pen-up events, number of strokes etc.) to be captured is prepared. Then a software application or a data collection tool which runs on handheld devices like Tablet PC is developed. Writers input the selected text through a GUI available in the data collection tool. The data collection tool captures the requisite information as the writer writes. As writers may not be at ease with the writing interface of the digitizing tablet and the electronic pen, they should be instructed to practice by writing several characters in the text input box in the GUI before the actual recording of characters are started. After a writer becomes familiar with the online hand writing tools and the environment, his/her actual data recording process should take place.

2.2.2 Datasets

This section briefly describes a few datasets of online handwriting available in Western and Indian Scripts.

2.2.2.1 Standard Datasets in Western Scripts

Some examples of datasets in the online domain are *Pen-Based Recognition of Handwritten Digits* dataset [3], *IRONOFF* dataset [4], *UJIpenchars* dataset [5] and *UNIPEN* dataset [6]. *Pen-Based Recognition of Handwritten Digits* dataset contains 250 samples of digits collected from 44 writers. The samples written by 30 writers are used for training, cross-validation and writer dependent testing, and the digits written by the other 14 are used for writer independent testing. *IRONOFF* contains a large number of isolated characters, digits, and cursive words written by French writers. This database has been designed so that, given an online point, it can be mapped at the correct location in the corresponding scanned image, and conversely, each offline pixel can be temporally indexed. *UJIpenchars* dataset of characters was created by collecting samples from 11 writers. Each writer contributed with Spanish letters (lower and uppercase) and digits. The total number of samples in this database is 1364. *UNIPEN* dataset contains online handwriting data from various alphabets (including Latin and Chinese), signatures and pen gestures.

2.2.2.2 *Standard Datasets in Indian Scripts*

Research for online handwriting recognition has also emerged for Indian scripts. Examples of few Indian Scripts where online handwriting recognition activities are being carried out include Tamil [7,8,9] Telugu [9,10], Bengali [11,12], Devanagari [10,13,14], Gurmukhi [15] and Assamese [16,17,18,19,20]. Standard datasets of online handwritings are also developed for few Indian scripts. But not all datasets of Indian scripts are known to be publicly available. Examples of few datasets of online handwritings in the context of Indian scripts are namely, HP Lab Tamil Dataset [21], HP Lab Telugu Dataset [22], Bangla Numeral Dataset [23], Devanagari Character Dataset [24] and Online Handwritten Assamese Characters dataset [25]. HP Lab Isolated Online Handwritten Tamil Character Dataset contains approximately 500 isolated samples each of 156 Tamil characters written by native Tamil writers. The data was collected using HP Tablet PCs and is in standard UNIPEN format. The data is available only for research use. HP Lab Isolated Online Handwritten Telugu Character Dataset contains approximately 270 samples each of 166 Telugu characters written by native Telugu writers. The data was collected using Acecad Digimemo electronic clipboard devices using the Digimemo-DCT application. The data is in standard UNIPEN format. The dataset is available only for research use. The present Bangla Numeral Dataset consists of 6000 samples of online handwritten Bangla numerals. In each class there are 600 samples. The whole set of samples is divided randomly into training and test sets consisting of 4000 and 2000 samples respectively. The dataset is available on request. Isolated Handwritten Devnagari Character Dataset contains approximately 270 samples each of 111 Devnagari characters written by over 100 native Hindi speakers. The data was collected using Acecad Digimemo electronic clipboard devices using the Digimemo-DCT application. The data is in standard UNIPEN format. The dataset is available only for research use. The Online Handwritten Assamese Characters dataset developed as part of research carried out for this dissertation, contains

8235 samples of online handwritten Assamese characters which include Assamese numerals, basic alphabetic characters and conjunct consonants (*Juktakkhors*). A discussion on Online Handwritten Assamese Characters dataset is included in Chapter 3 of this thesis.

2.3 Preprocessing

Preprocessing of online handwriting data is performed prior to the application of character recognition algorithm. Preprocessing usually deals with filtering, smoothing and different types of normalization operations applied to online handwritten data. The online handwritten data usually include noise. Noise originates from the limiting accuracy of the tablet, the digitizing process, erratic hand motion and the inaccuracy of pen-down indication [1]. Noises may be in the form of jitters or roughness and missing points in the strokes. The characters may also vary in terms of size, orientation, number of coordinate points and the range of horizontal and vertical coordinate values. Variability in number of points results in variable lengths of the characters. Similarly, the characters may be written at different regions of the writing pad in the tablet. Besides noise removal, preprocessing involves the normalization of size, orientation, length and region of writing. Preprocessing is an important step in character recognition system. The goal of preprocessing is to discard irrelevant information that may negatively affect the recognition [26]. The noise and other variations in online handwritten data complicate handwriting recognition.

2.3.1 Preprocessing Steps

This section briefly describes various preprocessing steps. The preprocessing steps which usually are applied in character recognition system are noise reduction which deals with removal of jitters and other irregularities like hook.

Smoothing operation removes jitters or roughness from the strokes. Jitters results from hardware problem or erratic hand motion. Jitters make the trace of the stroke angular. Moving average filters are used to smooth the strokes by reducing roughness or jitters [27]. Smoothing at a point *P* is performed by computing the average of the point positions in a specific neighbourhood of the *P*, where *P* is the center of the neighbourhood [2]. When the pen moves at high speed some intermediate points of the stroke may be missing. Interpolation is used to approximate the missing points. Jarger et al. [27] describes a method to interpolate missing points using Bezier curve. He reported only a minor impact of interpolation of missing point on the recognition rates. Hooks are imperfections at the beginning and at the end of a stroke. Hooks result from erratic hand motion and inaccuracies in the contact of pen and the writing surface at the time of pen-up and pen-down movements. These are small in size and have great angular variations. Planmondon et al. [2] describes a hook removal process where strokes are processed at their extremities to remove hooks from that portion of the stroke based on the thresholds on the length and the angular variation between the points. Size normalization is the process of removal of variation in size. Normalizing writing size is used to simplify the character recognition process. Characters of different sizes are normalized to a fixed size. Size normalization of online handwriting are described in [28], [29] and [30]. The numbers of points in characters are variable. The variable numbers of points of the strokes are normalized to a fixed number of points. Jaeger et al. [27] describes a method for the normalization of points. The variability of the number of recorded points depends on the velocity of writing and the hardware used. To normalize the number of points, the sequence of recorded points is replaced with a sequence of points having the same spatial distance [27]. For normalizing the number of points, filtering or interpolation is used to reduce or increase, respectively, the number of points involved [31].

2.4 Features of Online Handwritten Characters

Features should represent the characters properly. The efficiency of a character recognition system depends on relevant features. A large variety of features of online handwritten characters are available. The features of online handwriting can be categorized as geometrical features, structural or topological features, and statistical features.

2.4.1 Geometrical Features

Geometrical features represent the geometrical properties of the character. The coordinate points of strokes, direction of strokes, start point and end point information of strokes; curvature etc. can be categorized as geometrical features.

2.4.1.1 Resampled Horizontal and Vertical Coordinates

Two dimensional (x,y) coordinate points are the basic geometrical features of online handwritten characters. The online handwritten characters are normalized (or resampling) in preprocessing step to have equal number of points. The resampled horizontal (x) and vertical (y) coordinates are considered as features (pen coordinate features) for online handwritings [9,10,13, 32, 34].

2.4.1.2 Pen-down and Pen-up Positions

Pen-down and pen-up features are binary features. Pen-down feature indicates the start of writing and pen-up feature indicates the end of writing. A pen-down event consecutively followed by a pen-up event represents a single stroke of a character. Pen-up and pen-down information is captured as an integral part of data acquisition. A string of coordinates as a function of time is recorded along the pen trajectory during the pen movement over the surface of the sensitive screen. This facilitates to track the number of strokes and their order within a character [35, 36].

2.4.1.3 Writing Direction

The change of writing direction is regarded as the change of orientation of stroke from one pen position to the next pen position. The writing direction is similar for one particular character or digit, though its stroke order may be different [35]. The local writing direction at a point (x(t),y(t)) is described using the cosine and sine as described below [27, 38, 39]. The angles associated with this computation are shown in the Figure 2.4.

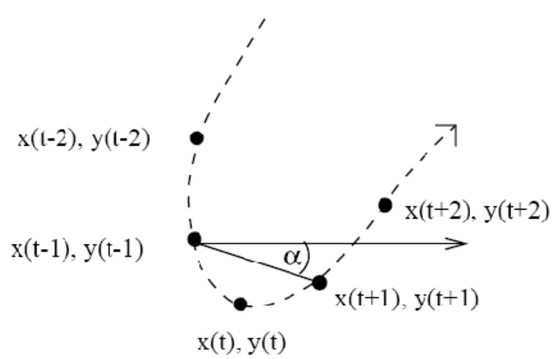


Figure 2.4. Writing Direction (Adopted from Jager et al.[27])

$\cos\alpha(t) = \dfrac{\Delta x(t)}{\Delta s(t)}$ and $\sin\alpha(t) = \dfrac{\Delta y(t)}{\Delta s(t)}$, where $\Delta s(t)$, $\Delta x(t)$ and $\Delta y(t)$ are defined as follows:

$$\Delta s(t) = \sqrt{\Delta x^2(t) + \Delta y^2(t)}$$
$$\Delta x(t) = x(t+1) - x(t-1)$$
$$\Delta y(t) = y(t+1) - y(t-1)$$

2.4.1.4 Curvature Feature

Curvature is the reciprocal of the radius of a circle touching and partially fitting the curve [27]. Curvature feature of an online handwritten character indicates the degree by which the trajectory of writing deviates from being straight. The curvature at a point (x(t),y(t)) can be implemented using the Cosine and Sine of the angle with the following series of points [27, 38, 39]

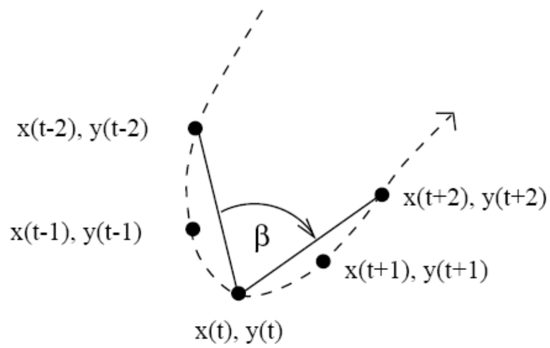


Figure 2.5. Curvature (Adopted from Jager et al.[27])

$(x(t-2), y(t-2))$, $(x(t), y(t))$, $(x(t+2), y(t+2))$. The angles associated with this computation are shown in the Figure 2.5. The angle $\beta(t)$ is given by $\beta(t) = \alpha(t+1) - \alpha(t-1)$. The curvature in terms of Cosine and Sine is defined as

$$\cos\beta(t) = \cos\alpha(t-1) \times \cos\alpha(t+1) + \sin\alpha(t-1) \times \sin\alpha(t+1)$$

$$\sin\beta(t) = \cos\alpha(t-1) \times \sin\alpha(t+1) - \sin\alpha(t-1) \times \cos\alpha(t+1)$$

where $\cos\alpha(t) = \dfrac{\Delta x(t)}{\Delta s(t)}$, $\sin\alpha(t) = \dfrac{\Delta y(t)}{\Delta s(t)}$ and $\Delta s(t)$, $\Delta x(t)$ and $\Delta y(t)$ are defined as $\Delta s(t) = \sqrt{\Delta x^2(t) + \Delta y^2(t)}$, $\Delta x(t) = x(t+1) - x(t-1)$ and $\Delta y(t) = y(t+1) - y(t-1)$.

2.4.1.5. Eight-directional Feature

Eight-directional feature of online handwriting is based on Free-man Code [40]. Starting from first pen-down event, direction in which the pen tip moves is recorded along the directions, namely 0,1,2,3,4,5,6 and 7 as shown in the Figure 2.6) and outputs a feature value representing the direction of this movement. The eight-directional feature is mostly used in Chinese, Japanese, and Korean characters, where a large number of symbols is used to complete a character, many standard strokes/line segments are defined as the basic components of the character. In these scripts, typically, strokes as the directional arrows are of eight types, coded

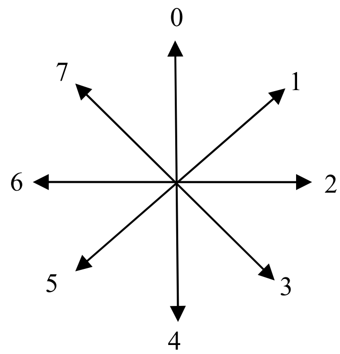


Figure 2.6. Freeman code

from 1-8. Apart from the mentioned scripts (Japanese, Chinese, and Korean) also in roman, the directional codes are used as feature [36]. Variants of eight-directional features are used in [41, 42, 43].

2.4.2 *Structural or Topological Features*

Structural features are based on topological properties of the character, such as aspect ratio, cross points or junction points, loops, cusps, hook, isolated dots etc.

2.4.2.1 *Junction Point*

Junctions are the intersection of strokes. Figure 2.7 shows a junction point. Junction points are obtained when two or more strokes intersect. Junction points are present in multi-stroke characters only. This feature happens to be missing in cursive writing, since most of the characters are written using single stroke in cursive writing. The extraction of junction point is performed in [44, 45, 46].

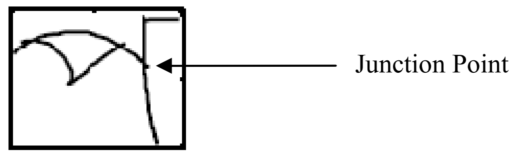


Figure 2.7. Junction point

2.4.2.2 Loop

Loops are the small circular parts attached to the main body of the character. Loops result from self intersecting strokes. When a stroke intersects itself at some point then a loop is created at that point. Figure 2.8 shows a loop. Loop extraction techniques for online Thai character recognition are found in [37, 56]. Loop feature is also used in [10, 45, 46, 47].

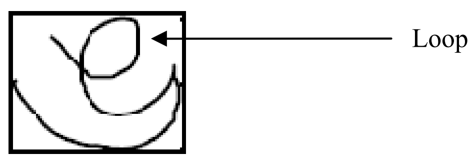


Figure 2.8. Loop

2.4.2.3 Cusp

Cusps are points of sharp directional change. S Krasnik [48] defines cusp points as high-curvature points in the input. A cusp is a point at which two branches of a curve meet such that both the tangents one for each branch coincide. Figure 2.9 shows a cusp. Cusp feature is used in an online handwriting recognition explained in [7, 46].

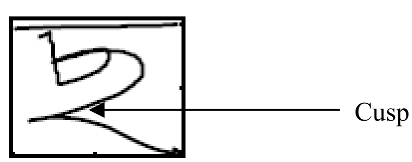


Figure 2.9. Cusp

2.4.2.4 *Dot*

A dot appears to be a very small stroke and it is an isolated stroke. Figure 2.10 shows a dot. The detection of dot feature from online handwritten characters depends on stroke length, stroke direction, stroke position and nature of points in the stroke. The use of dot feature is mostly found in Arabic characters [47, 49]. The extraction of dot feature is also performed in [7, 45].

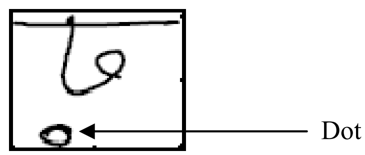


Figure 2.10. Dot Feature

2.4.2.5 *Head Line*

Head line is the horizontal stroke at the upper zone of the character. Figure 2.11 shows a head line. The horizontal stroke is drawn on top of all associated character which is also referred *Shirorekha*. The head line vertically separates a character from its neighbours [13]. Roman character set does have the head line feature and it exists mostly in case of Indian scripts. [13, 45] used headline as a feature in online character recognition.

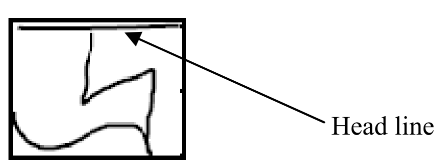


Figure 2.11. Head Line Feature

2.4.2.6 *Vertical Line*

The use of vertical line feature for online handwritten character recognition found in [50, 51]. Figure 2.12 shows a vertical line. The feature values extracted from vertical lines are namely the numbers and positions of the vertical lines in the characters [52].

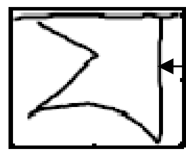


Figure 2.12. Vertical Line Feature

2.4.3 *Statistical Features*

Statistical features are numerical measures derived from the constituent points of the strokes. Some examples of statistical feature include normalized horizontal coordinates, normalized vertical coordinates, zoning, distance, number of strokes, character length, aspect ratio etc.

2.4.3.1 *Normalized horizontal and vertical coordinates*

Normalized pen coordinates are used as features described in [38, 53, 54, 55]. Normalized horizontal and vertical coordinates are given by $\bar{x}_i = \frac{x_i - \mu_x}{\sigma_y}$ and $\bar{y}_i = \frac{y_i - \mu_y}{\sigma_y}$, where (x_i, y_i) is the point in the online handwritten stroke having horizontal coordinates x_i and vertical coordinates y_i, normalization is performed with mean μ_x and μ_y of horizontal x and y coordinates and σ_y [54].

2.4.3.2 Zone

The bounding rectangle of a character is divided into M×N zones of equal intervals. Each zone contains distinct information. Stroke information, for instance number of points in each zone is used as features. The use of zoning information as feature is explained in [35]. F. Deborah et al. [44] has also proposed a zoning algorithm.

2.4.3.3 Mean of horizontal and mean of vertical coordinates

Mean of horizontal coordinates (x) and that of vertical coordinates (y) of the pen coordinate point (x,y) are used as features for online handwriting recognition in [13].

2.4.3.4 Standard Deviation

In the context of online handwriting recognition standard deviation of horizontal coordinates (x) and that of vertical coordinates (y) of the pen coordinate point (x,y) is used as a feature. Standard deviation feature is used in [55, 57].

2.4.3.5 Number of strokes

The number of constituent stroke in an online handwritten character is derived from the pen-down events. The number of constituent strokes in an online handwritten character is equal to the number of occurrences of pen-down operations [35].

2.4.3.6 *Stroke length*

The lengths of the individual strokes of an online handwritten character are computed and used as a feature. The length of a stroke is defined as the number of points in the stroke [36]. The usage of stroke length as feature is defined in [34]. The eight-directional feature of a stroke conveys more information if it is supplemented by the length of the stroke as explained in [36, 58]

2.4.3.7 *Aspect Ratio*

The aspect ratio of an online handwritten character is defined as the ratio of the height to the width of the character. The use of aspect ratio as feature is used in [34, 59, 60]. When long shaped characters are converted to square shape, the conventional normalization method may distort the shape of the character excessively. C. L. Liu et al [61] proposed a new method for normalization of character called aspect ratio adaptive normalization (ARAN) by incorporating aspect ratio into normalization procedure to control the aspect ratio of the normalized image.

2.4.3.8 *Derivative feature*

The first derivative features from pen coordinates x and y are computed using the equations $x'_j = \frac{\sum_{k=1}^{2} k(x_{j+k} - x_{j-k})}{2\sum_{k=1}^{2} k^2}$ and $y'_j = \frac{\sum_{k=1}^{2} k(y_{j+k} - y_{j-k})}{2\sum_{k=1}^{2} k^2}$, where the range of k defines the window size which determines the number of neighbor points involved in the computation [62]. The Second derivatives x'' and y'' are computed in the same way as the first derivatives by replacing x and y by x' and y' [62]. Derivative features are also introduced in [9,63, 64].

2.4.3.9 Distance

Distance measures are used as features for the recognition of online written characters. F. Bhattacharyya et al. [65] used a distance feature by incorporating the horizontal and vertical distances between terminal and the final points in a stroke. Wujiahemaiti Simayi et al. [66] introduced a distance feature called center distance feature [CDF] for online handwriting Uyghur character. Askar Hamdulla et al. [67] introduced modified center distance feature [MCDF] by incorporating stroke number feature, additional part's location feature, shape feature, bottom-up and left-right density feature into CDF.

2.5 Qualitative Representation of Planar Outlines

This section explains the representation of planar shapes in qualitative domain. The shape of an object can be described in both quantitative and qualitative ways. The quantitative representation of a planar shape involves a set of mathematical functions of plane coordinates and features described as in previous section. If the shape is more complex, it is difficult to find a mathematical function for the curve describing the outline or boundary of the shape [68]. Qualitative representation makes use of symbolic schemes for describing shapes. Qualitative representation of describing shapes is considered preferable to quantitative techniques. This is because qualitative techniques can deal with abstract or complex shapes more efficiently than purely quantitative models [69]. Descriptions about qualitative representation of shape can be found in [70, 71, 72].

2.5.1 What is Qualitative Reasoning?

Qualitative Representation of shape draws inspiration from Qualitative Reasoning and more specifically Qualitative Spatial reasoning. Qualitative Reasoning (QR) is an area of research which is a subfield of Artificial

Intelligence. Qualitative reasoning automates reasoning about space, time and quantity, which are the continuous aspects of the physical world. Qualitative reasoning uses qualitative rather than quantitative information. Instead of using precise numerical values or quantities, qualitative reasoning uses qualitative values (e.g., high, low, left, right, acute angle, obtuse angle, etc.). One tries to represent knowledge using qualitative relationships between entities or qualitative categories of numerical values [73]. For instance, all quantitative values of angles less than ninety degree can be represented using the qualitative categorical term "acute angle". Qualitative approach is very close to how human beings represent and reason about commonsense knowledge in real life. Qualitative spatial reasoning is a sub-field of qualitative reasoning which is concerned with spatial aspects of qualitative representation of objects (e.g. orientation, distance, direction, topology, shape, size, etc.). Some of the areas where QSR has been applied are Geographical Information System (GIS), robotics, natural language processing, computer vision etc.

2.5.2 Qualitative Orientation

Qualitative orientation involves representation and reasoning about orientation of spatial entities in qualitative domain. Orientation of spatial entities with respect to other spatial entities is usually given in terms of a qualitative category rather than using a numerical expression. Orientation of spatial entities depends on the located object, the reference object, and the frame of reference which can be specified either by a third object or by a given direction. Most approaches to qualitatively dealing with orientation are based on points as the basic spatial entities and consider only two-dimensional space [73]. Here we briefly highlight a qualitative orientation model QDA_8 proposed in [74]. In this orientation model four qualitative direction relations are introduced namely, $Same$, $Opposite$, LR and RL. In $ASameB$ relation, the objects are directed in the same direction. $LeftToRight$ relation (LR) means that one object is directed in a left-to-right orientation with respect to the other in $RightToLeft$ (RL). If an object moves in LR direction with respect a reference object, its course of motion is from the left to the right with respect to the reference object and intersects the course of motion of the reference object at 90 degrees. A span of 45 degrees counterclockwise is denoted by a + and the same in clockwise direction is denoted by a $-$. If the direction of the object makes an angle less than or equal to 45 degrees anticlockwise with the direction of the reference object, the resulting relation is $Same+$ and in the counterclockwise case, it is $Same-$. The wheel shown in Figure 2.13 divides 360 degrees into eight regions, each having a span of 45 degrees. This model is referred to as QDA_8 to express the fact that it is Qualitative Direction Algebra (QDA) with granularity equal to 8.

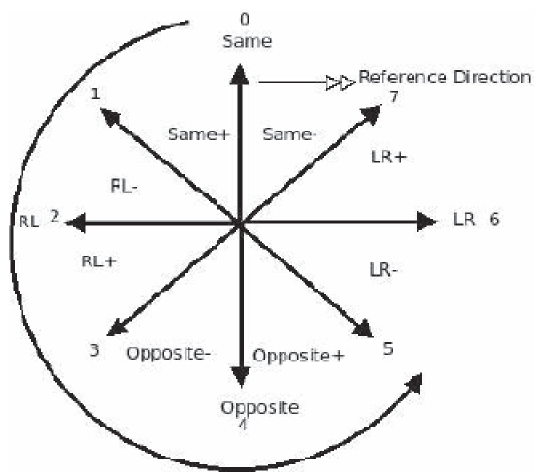


Figure 2.13. Direction Relations in terms of orientation angles (Adopted from ISDA-R Baruah & S M Hazarika [74])

Similarly, the relations namely, $Opposite+$, $Opposite-$, $LR+$, $LR-$, $RL+$ and $RL-$ are derived. Changes in direction are noticed after an interval of 45 degrees from one direction to the next. In Table 2.1, these relations are listed along with direction regions in terms of angles of the bounding direction lines.

Table 2.1. Drection Relations (Adopted from ISDA-R Baruah & S M Hazarika [74])

Sl. No.	Base Relation	Angle Range	Converse of Base Relation	Sl. No.	Direction Relation	Angle Range	Converse of Base Relation
1	$Same$	$[0, 0]$	$Same$	7	lr	$[270, 270]$	rl
2	$Same+$	$]0, 45]$	$Same-$	8	$lr+$	$]270, 315]$	$rl-$
3	$Same-$	$]315, 360[$	$Same+$	9	$lr-$	$]225, 270[$	$rl+$
4	$Opposite$	$[180, 180]$	$Opposite$	10	rl	$[90, 90]$	lr
5	$Opposite+$	$]180, 225]$	$Opposite-$	11	$rl+$	$]90, 135]$	$lr-$
6	$Opposite-$	$]135, 180[$	$Opposite+$	12	$rl-$	$]45, 90[$	$lr+$

2.5.3 *Qualitative Outlines*

Qualitative outlines of shapes involve representation and reasoning about outline or boundary of shapes in qualitative domain [69,75]. Shape is one of the most important characteristics of an object, and equally complex to describe qualitatively. Even though, with the interest in qualitative spatial reasoning, much research has focussed on spatial attributes such as position, orientation, etc. through numerous topological relationships, explicit qualitative shape representation has remained a less explored area. In a purely topological theory very limited statements can be made about the shape of a region: whether it has holes, or interior voids, or whether it is one piece or not. However, for an explicit qualitative shape description one needs to go beyond topology, introducing some kind of shape primitives whilst still retaining a qualitative representation. The existing approaches to qualitative shape description can be classified into the following categories.

- Boundary representations, which primarily describe the boundary of an object.
- Axial representations, which represent the interior (e.g. symmetry based techniques)
- Shape abstractions, which constrain the possible shape of a region within a bounding box or a convex hull.
- Synthetic, which uses a set of primitive shapes to generate a complex shape.

Examples of approaches which work by describing the boundary of an object include those that classify the sequence of different types of boundary segments [76] or by describing the sequence of different types of curvature extrema [77] along its contour. Explicit representation of shape using boundary representation has been undertaken by [75]. In the following subsections we present brief discussions on three qualitative descriptions of boundary representations introduced in [76], [77] and [75].

2.5.3.1 Codon Theory of Richard and Hoffman

Contour cordons or simply codons are simple shape primitives introduced by Hoffman & Richards (1982) [76]. Codons can be used as shape descriptors for describing planar curves. A codon can be defined as a small a curve segment that is characterized by curvature minima. A codon can contain zero point of zero curvature, one point of zero curvature or two points of zero curvature. The four basic types of codon as described by Hoffman & Richards are symbolized as 0, 1^-, 1^+ and 2 depending on the number of zero curvature points present in segment of curve. Type 0 codon contains no zero curvature point, type 1^- codon contains one point of zero curvature which occurs before the point of maximum curvature, type 1^+ codon contains one point of zero curvature which occurs after the point of maximum curvature and type 2 codon contains two points of zero curvature. Figure 2.14 shows all four types of codons where dots in the curve indicate zeros of curvature and slashes denote curvature minima.

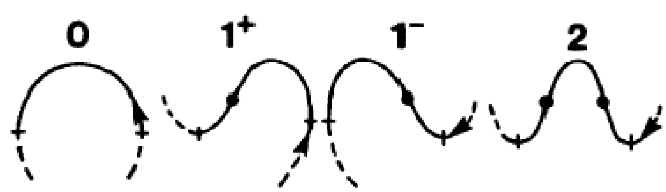


Figure 2.14. Contour Codons (Adopted from Hoffman & Richards [76])

A curve is divided into segments using the curvature minima. Both open and closed planar curves can be represented by strings of contour codons. Moreover, only certain codon joins in pairwise connections are allowable [76]. The allowable pairwise connections of codons are shown in Table 2.2, rows and columns are labeled by codon types where a tick mark indicates allowable codon joins and a cross mark indicates a non allowable codon join. An example of a shape represented in terms of string of contour codons is shown in Figure 2.15.

Table 2.2. Allowable Codon Joins

	0	1^-	1^+	2
0	√	√	√	√
1^-	√	×	×	√
1^+	√	√	×	√
2	√	√	×	√

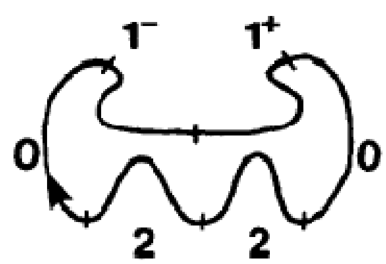


Figure 2.15. Codon String: 01^-1^+022 (Adopted from Hoffman & Richards [76])

2.5.3.2 Leyton's Process Grammar for Shape

Curves can be divided into smaller curves based on the symmetry axis of curves and can be described in terms of curvature extrema [77]. Four types of curvature extrema defined by Leyton [74] are namely, M+, m-, m+ and M-. All these four types of curvature extrema are shown in Figure 2.16. Among these four curvature extrema, M+ and m- are the sharpest curvature extrema having exactly the same shape. However, M+ differs from m- in the sense that, in M+, the solid portion (shaded) is on the inside, and, in m-, the solid portion (shaded) is on the outside. This characteristic make them figure/ground reversals of each other. The other two extrema m+ and M- are also figure/ground reversals of each other. In this case, the extrema m+ and M- are the flattest points on the respective curves. In Figure 2.17 a planar curve is represented in terms of six curvature extrema, where arrows denote axes of symmetry.

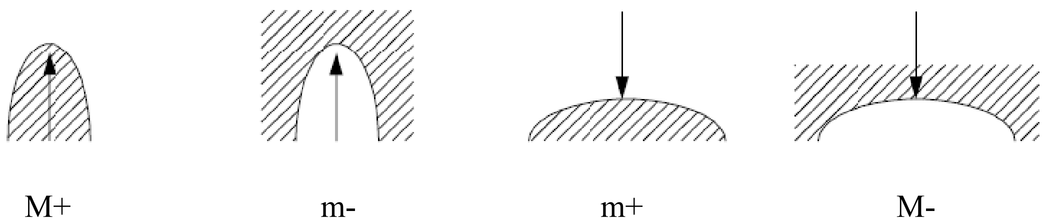


Figure 2.16. The Four Types of Extrema (Adopted from Leyton [77])

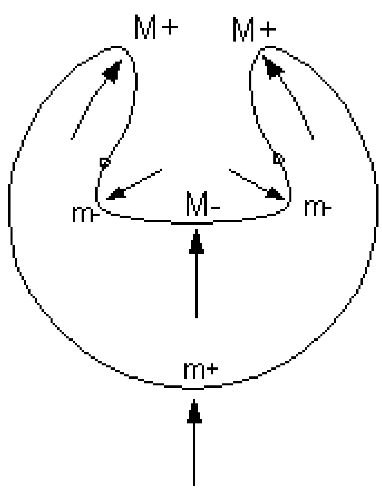


Figure 2.17. A planar curve with six extrema (Adopted from Leyton [77])

2.5.3.3 Qualitative Outline Theory of Meathrel and Galton

A scheme for the qualitative description of two-dimensional outlines is introduced in [75]. This qualitative description of the planar outline is in terms of curvature components defined in the system. The complete listing of all the qualitative curvatures introduced is shown in Table 2.3. A sequence of qualitative curvature types ordered in specific way defines the complete shape of the planar outline.

Table 2.3. Types of Qualitative Curvatures [75]

Qualitative Curvature Type	Description
/	Straight line segment
$\supset$	Convex curve segment
$\subset$	Concave curve segment
$>$	Outward pointing angle
$<$	Inward pointing angle
$\succ$	Outward pointing cusp
$\prec$	Inward pointing cusp

A shape is illustrated in Figure 2.18 which consists of all the seven qualitative curvatures listed in the Table 2.3. Each curvature type appears once in the figure.

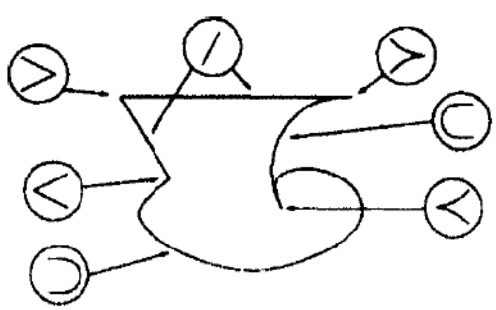


Figure 2.18. Seven Qualitative Curvature Type [75]

The qualitative curvature types are grouped in many of ways. The group can be considered as representing length (*linelike* elements) of an outline, while the group of the remaining elements $(>, <, \succ, \prec)$ angular changes of the outline (*pointlike* elements). Another way of making group is on the basis of *outward* and *inward* curvature types. The outward group is $(\supset, >, \succ)$ and the inward group is $(\subset, <, \prec)$. In this scheme of grouping the curvature type / (straight line) does not belong to any group.

2.5.4 *Approaches to Curvature Components*

This section provides a brief discussion on approaches to curvature components. A number of qualitative shape descriptions have been introduced in recent years. A few approaches to qualitative curvature components are highlighted in the previous section on qualitative outline theory. Curvature components associate both qualitative and quantitative considerations [69, 75]. In certain situations qualitative considerations alone is not sufficient to make distinction among the objects. Quantitative consideration supplements the qualitative approach, which is explained in the following subsection.

2.5.4.1 *Quantitative Consideration of Curvature Components*

In a qualitative representation system, it is possible that single representation corresponds to many different objects. Objects treated identically in the system may however differ noticeably corresponding to features which are beyond the accessibility of the system. An example of this is shown in Figure 2.19, where five different shapes are of the qualitative outline type $\supset\prec\supset\prec$. The figures are not purely quantitatively different, but they are quantitative in the sense that they are to use relative lengths and curvatures of the line-like segments. For a qualitative representation with curvature types discussed in sub-section 2.5.3.3, it happens to include each of the *quantitative* features. The exact expression of *quantitative* features requires measurements based on the use of real-number. Without quantitative considerations of curvatures, the system would be unable to discriminate between these different outlines.

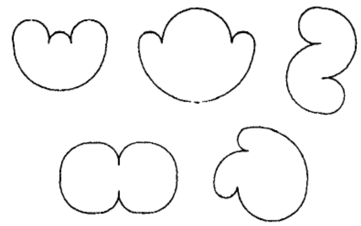


Figure 2.19. Distinct patterns of the outline types [75]

The qualitative representations can be extended by adding more information to make distinction among these shapes. For example, the index U can be used to annotate the symbol $\prec$ to indicate "upward pointing" property of the symbol. Similarly, indices D, L and *R* can be used to annotate the symbol $\prec$ to indicate "downward pointing", "left pointing" and "right pointing" properties respectively. As per this scheme, first two outlines in Figure 2.17 are represented as $\supset\prec_{D}\supset\prec_{D}$, and the others as $\supset\prec_{L}\supset\prec_{R}$, $\supset\prec_{D}\supset\prec_{U}$ and $\supset\prec_{R}\supset\prec_{U}$. These expressions represent the shapes qualitatively which allow considerable flexibility and in this scheme they certainly capture the essential features of the visual appearance of the outlines. Using the same scheme, the line-like elements $\supset$ and $\subset$ can be annotated by indices which denote their relative lengths (a quantitative measure). This can be done by adopting "short" (S), "medium" (M) and "long" (L). Similarly, the types of angle $<$ and $>$ can be annotated using an indication of the size of the angle and the direction it is pointing which are essentially quantitative considerations.

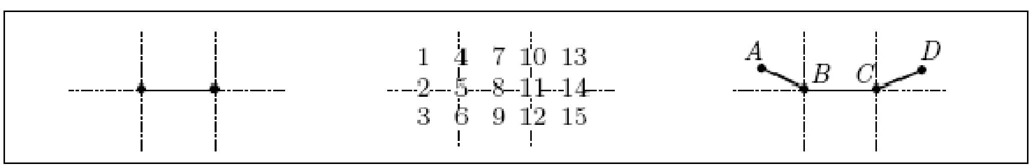
2.5.4.2 Qualitative Curvature Components

This section discusses two instances of qualitative representations of planar shapes with qualitative curvature components or qualitative curvature types.

These two representations are namely, Tripartite Line Tracks (TLTs) [78] and Primitive Curve Tokens (PCTs) [69].

2.5.4.2.1 Tripartite Line Tracks

This sub-section introduces *Tripartite Line Tracks* [78] as a qualitative curvature component. The concept of *Tripartite Line Tracks* was introduced to describe two-dimensional boundaries of polygonal shapes in a qualitative way. This qualitative description involves the use of shape primitives. These primitives are tracks of three lines which are described by the orientation grid. The orientation grid partitions the plane into six regions. Figure 2.20 (a) shows an orientation grid (dotted lines outline those six regions). These three lines are in particular suitable for the description of two dimensional shape primitives.



(a) (b) (c)

Figure 2.20. (a) The orientation grid introduced by one continuously drawn line, (b) Fifteen distinguishable positions, (c) A line track (*AB,BC,CD*) consisting of three connected lines [78]

For the construction shape primitives only general positions are considered which are the six positions not lying on the orientation grid: 1, 3, 7, 9, 13, and 15 in Figure 2.18.(b). The other nine positions called singular positions directly lie on the orientation grid. Since, both endpoints of one line track can lie in six different general positions; therefore, there exist $6^2 = 36$ different relations. This is outlined in Figure 2.21. These line tracks are known as *Tripartite Line Tracks* (*TLT*). *TLT* (*i*) accesses the i^{th} relation. Indices distinguish specific *TLT* -subsets. *TLT* provides a local feature scheme for polygons. It characterizes polygons by considering the arrangement of a polygonal line together with its predecessor and successor, i.e. by considering its local context. Figure 2.19 shows these relations, which are oriented from left to right; *TLT* (13) and *TLT* (24) can occur in the context of self-intersecting polygons (complex polygons) only. A polygon is locally characterized by determining for each line segment the corresponding *TLT*. When three connected lines are considered it is possible to distinguish whether the two outer lines are on the same side with respect to the medial line or not, and which combinations of acute and obtuse angles exist.

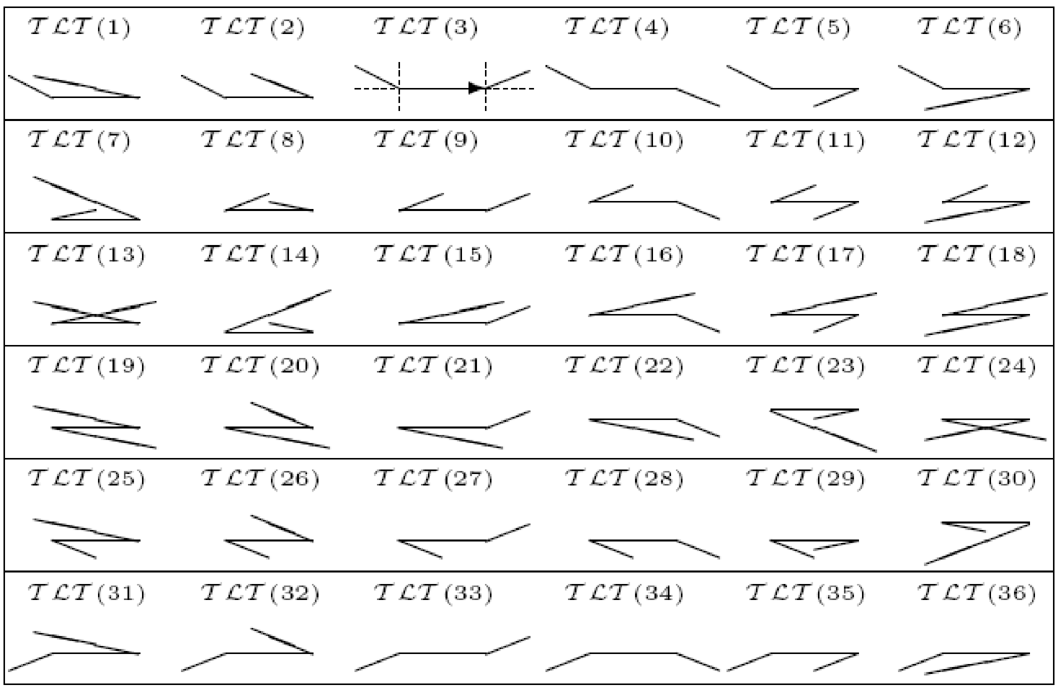


Figure 2.21. Distinguishable Classes of Line Track Arrangements with Three Connected Lines[78]

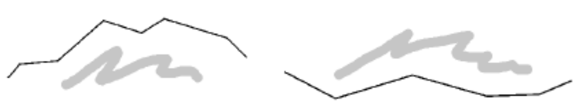


Figure 2.22. TLT representations of polygon sides [78]

Figure 2.22 shows typical examples which can be characterized by patterns of concave *TLT* -relations, such as chains of *TLT* (4, 33).

2.5.4.2.2 Primitive Curve Tokens

Curve tokens can be defined as curvature types which describe boundaries qualitatively. Primitive curve tokens (PCTs) of higher-level are specified by grouping together strings of atomic tokens. PCTs correspond to localized curve features of greater abstraction [69]. Schemes which represent curves by strings of tokens that correspond to local features encountered whilst a curve is traversed are called *local feature schemes*. In [69], a set of *atomic curve tokens* is derived based on tangent bearing and curvature, which can be used to specify the PCTs.

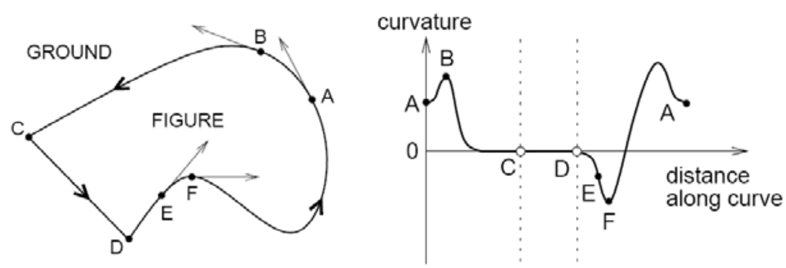


Figure 2.23. An example outline and its curvature plot.[69]

An instance of outline and the corresponding curvature plot are shown in the Figure 2.23. The tangent is rotating anticlockwise at points A and B, which indicates a positive curvature segment. At points E and F the tangent indicates negative curvature. The curvature is undefined at points C and D. A local feature scheme is used to describe a curve through a string of tokens, $t_1, t_2, \ldots\ldots, t_n$ that symbolize significant features of the curve. Token strings representing outlines can be interpreted as cyclic or ring. Even though a given outline correspondence exist between a given outline and a unique "token-ring", a number of corresponding token-strings take place. For example, an outline which consists of distinctive features, namely A, B, C, and D (in the order from A to D), has equivalent description with one arbitrary string from the set four strings namely, ABCD, BCDA, CDAB, and DABC [69]. A set of tokens of *base-level* is derived by combining discretizations of curvature and tangent bearing. At arbitrary point in a curve, the curvature can be negative (-), zero (0), positive (+) or undefined (*U*). Therefore, for curvature, the discrete quantity space {+, 0,-, *U*} is used. For tangent bearing the space {*D, U*} is used, since at some points along a curve the tangent may undefined (*U*) and it is defined (*D*) at all other points. At an arbitrary point p in a curve, the *curve state* can be specified by the pair $<b_p, c_p>$, where c_p, b_p denote the qualitative values of the curvature and tangent bearing a *p*, respectively. The set of base-level tokens are obtained at an interval or at a singular point. Table 2.4 presents the set consisting of six base-level tokens denoted by *BaseTok*. Here, there are three *interval tokens*, which represent curve states which can continue in segments of curve ($\underline{P}$, $\underline{Z}$, and $\underline{N}$) and three *point tokens*, which represent curve states which are held at singular points (Z, U_c, and U_b). Internal tokens are represented by underlined letters.

Table 2.4. The six tokens of BaseTok [69]

	Tokens	
b c	*Interval*	*Point*
D $+$	$\underline{P}$	
D 0	$\underline{Z}$	Z
D $-$	$\underline{N}$	
D U		U_c
U U		U_b

An example of representations of three different shapes using the tokens of *BaseTok* is shown in the Figure 2.24. The corresponding token-string descriptions are also shown.

Shape	*Description*
circle	$\underline{P}$
square	$\underline{Z}\ U_b\ \underline{Z}\ U_b\ \underline{Z}\ U_b\ \underline{Z}\ U_b$
sausage	$\underline{P}\ Z\ \underline{N}\ Z$

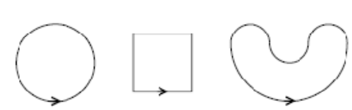


Figure 2.24. Describing three different shapes with *BaseTok* [69]

The atomic curve tokens are the extensions of BaseTok. All instances of the category of planar curves can have the representation with the set of curve tokens of base-level. But due to its limited discriminating capacity, the BaseTok tokens are extended by adding the rate at which curvature changes with distance along the curve (c') to the existing components of the curve state. Hence, the curve state at arbitrary point p in a curve can be given by the triple $<b_p, c_p, c'_p>$. The set of extended base-level tokens are obtained at an interval or at a singular point or both over an interval of points and hold at a singular point. Table 2.5 lists the valid composite combinations of fourteen numbers and Figure 2.25 shows the seventeen tokens (state labels) which are used to represent the allowed interval and point interpretations. The set of atomic *curve tokens* (also called "atoms") are collectively made up of those tokens. Atoms are also referred to as AToks. The curves, otherwise considered identical under BaseTok are distinguished by the elements of ATok. This is because of the fact that greater level of discriminatory ability is possessed by the atomic tokens.

Table 2.5. Curve state combinations [69]

	Tokens	
$b\ \ c\ \ c'$	*Interval*	*Point*
$D\ +\ +$	$\underline{P^+}$	
$D\ +\ 0$	$\underline{P^0}$	P^0
$D\ +\ -$	$\underline{P^-}$	
$D\ +\ U$		P^U
$D\ 0\ +$		Z^+
$D\ 0\ 0$	$\underline{Z^0}$	Z^0
$D\ 0\ -$		Z^-
$D\ 0\ U$		Z^U
$D\ -\ +$	$\underline{N^+}$	
$D\ \ \ \ 0$	$\underline{N^0}$	N^0
$D\ -\ -$	$\underline{N^-}$	
$D\ -\ U$		N^U
$D\ U\ U$		U_c
$U\ U\ U$		U_b

$$ATok = \{\underline{P^+}, \underline{P^0}, P^0, \underline{P^-}, P^U, Z^+, \underline{Z^0}, Z^0, Z^-, Z^U, \underline{N^+}, \underline{N^0}, N^0, \underline{N^-}, N^U, U_c, U_b\}$$

Figure 2.25. The seventeen tokens of *ATok* [69]

PCTs can represent local curve features which correspond to a sequence of one or more curve states. PCTs are higher level curve tokens specified by the atoms of ATok. A sequence of curve-state is a valid string of atomic tokens and it marks out a route through ATok. Consequently, local curve features can be defined by the curve state sequences. Strings of length one define simple features, e.g. straight line segment can be defined by $\underline{Z}^0$. Strings of greater length can define more detailed features. A finite set consisting of curve-state sequences specifies a *PCT* T where T = $\{S_1, S_2, \ldots, S_n\}$, such that:

(i) Each S_i has the form *lc[id]tc*, where *id* represents an *identity* substring and *lc* and *tc* denote leading and trailing *context* substrings respectively. The identity substring identifies T with occurrences of *id*, while *lc* and *tc* denote the exact *context* where T is to be so identified.

(ii) T can be interval or point token. The type of T is determined by the identity substrings of T such that T is a point token if each *id* is a single point atom then, whereas T is an interval token if each *id* contains an interval atom.

Figure 2.26 illustrates an example of a PCT which identifies straight lines bounded by arcs of the same curvature sign.

$$T = \{\ \underline{P^0}U_c[\underline{Z^0}]U_c\underline{P^0},\ \underline{N^0}U_c[\underline{Z^0}]U_c\underline{N^0}\ \}$$

Figure 2.26. Illustartion of a PCT [69]

2.6 Online Handwriting Classification Methods

This section briefly describes the various classification methods used for Online Handwriting Recognition. These methods are based on Deformable Template

Matching or Elastic Matching, Statistical, Syntactic or Structural and Neural Networks approaches [45, 79].

2.6.1 Deformable Template Matching or Elastic Matching

Matching operation in pattern recognition is used to determine the similarity between two shapes of the same type. Template matching is a simple approach to pattern recognition. In template matching, a template which is typically a 2D shape or a prototype of the pattern to be recognized is available. The pattern to be recognized is matched against the stored template while considering all allowable geometrical transformations like translation, rotation and scale changes. The similarity measure is optimized based on the available training set. Usually, the template itself is learned from the training set. Even though template matching is computationally demanding, the availability of faster processors has now made this approach more feasible [80]. But it has some disadvantages in the case of matching of deformable shapes. For example it will fail if the patterns are distorted because of intra-class variations among the patterns. Deformable template models (also called Elastic Matching) can be used to match patterns when the deformation cannot be easily modeled or explained directly. Elastic matching can be defined as a process related to string matching [80]. In this process two sequences are compared together where a distance function is used to measure the distance between two different items of the given sequences. Elastic Matching is extensively used in the area of handwritten character recognition [81]. The advantage of Elastic Matching is that it works very well for writer-dependent data, since writer-dependent data does not require a relatively large amount of training data. The disadvantage is that it does not generalize well for writer-independent tasks and classification time grows linearly with the number of training examples [82]. Sridhar et al. [83] proposed a technique for the recognition of online handwritten characters based on elastic matching and dynamic time warping (DTW). A novel generative classifier called Active-DTW was described which combined Active Shape Models with Elastic Matching. Experimental results on IRONOFF dataset showed that the Active-DTW classifier can be substantially promising as a generative classifier. Madhavanath et al. [84] compares elastic matching schemes in case of writer dependent on-line handwriting recognition for isolated Tamil characters. Dominant point co-ordinates, quantized slope values and preprocessed x-y co-ordinates are used as features. A comparison of seven schemes based on these three features and DTW distance measure is presented with respect to recognition accuracy, recognition speed, and number of training templates. Error analysis and possible grouping strategies are also presented. Sharma et al. [15] present online handwritten Gurmukhi character recognition system based on elastic matching technique. Characters are recognized in two stages. In the first stage the strokes are recognized. In second stage an evaluation of character is performed on the basis of recognized strokes. A set of 41 Gurmukhi characters are used from 60 writers. The recognition rate of 90.08% is obtained.

2.6.2 Statistical Approach

In statistical classification approaches a shape is described by a fixed number of features. Features define a multidimensional representation space called the feature space in which different classes of shapes are described. That is, each pattern has the representation in terms of f features. In this representation each pattern is viewed as a point in f-dimensional feature space. The objective is to identify the features that allow pattern vectors belonging to different classes (or categories) to occupy disjoint and compact regions in the f-dimensional feature space [80]. The accuracy of the separation of patterns from different categories determines the effectiveness of the feature set. For a set of training patterns belonging to each class, decision boundaries in the feature space are established. Each decision boundary separates patterns belonging to different classes. In the statistical decision theoretic approach, the decision boundaries are determined by the probability distributions of the patterns belonging to each class, which must either be specified or learned. In discriminant analysis-based approach to classification, first a parametric form of the decision boundary (e.g., linear or quadratic) is specified and then the "best" decision boundary of the specified form is found based on the classification of training patterns. Examples of commonly used statistical techniques are Support Vector Machine (SVM) and Hidden Markov Model (HMM).

2.6.2.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a statistical technique for data classification [85]. The goal of SVM is to produce a model (based on the training data), which predicts the target values of the test data given only the test data attributes.

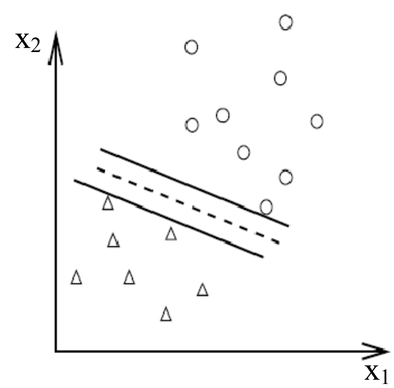


Figure 2.27. Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

Given a training set of instance-label pairs (x_i, y_i), $i = 1,...........,l$ where $x_i \in R^n$ and $y \in \{1,-1\}^l$, the SVMs require that a solution is found for the following optimization problem:

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$$\text{subject to} \quad y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0$$

Here, the training vectors x_i are mapped into a higher dimensional space by the function ϕ. SVM finds a linear separating hyper plane with the maximal margin in this higher dimensional space. Figure 2.27 shows a maximum-margin hyperplane and margins for an SVM trained with samples from two classes. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is called the kernel function. The three types of kernels that are used the most are as follows:

- Linear: $K(x_i, x_j) = x_i^T x_j$
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0$

Here, γ, r and d are the kernel parameters. In this experiment, we have used the SVM with the linear, polynomial, and RBF kernels.

The application of SVM is found to be very successful in online handwriting recognition [54]. Bhalman et al. [54] proposed a online handwriting recognition system based on combining dynamic time warping and SVM by establishing a new SVM kernel and this method is proven to be superior than the common HMM techniques. Jain et al. [86] proposed a procedure for recognizing online handwritten scripts from a document containing six scripts. Several classifiers are used and it has been found that SVM based classifier gave best performance. Hybridization of SVM and HMM models are also possible as suggested by Ahmed at al.[39]. In their work it has been reflected that SVM performs better than Multi Layer Perception and Time Delay Neural Network. SVM based online handwriting recognition system for Devanagari and Telugu character is proposed by Swethalakshmi et al [10]. Ramakrishnan et al. [87] proposed an online handwriting recognition method based on SVM with radial basis function (RBF) kernel. They investigated the efficiency of using global features alone (discrete Fourier transforms (DFT), discrete cosine transforms (DCT), local features alone (preprocessed (x, y) coordinates) and a combination of both global and local features. The classifier was trained and tested on the Tamil hand-written character recognition dataset. They have obtained more than 95% accuracy on the test dataset. Further, they have used a combination of global and local features on a publicly available database of Indo-Arabic numerals and obtained an accuracy of more than 98%. Namboodiri et al.[88] presented a SVM based generic design for development of online handwriting recognizers for Indian languages, and demonstrated its effectiveness with two Indian languages, namely, Malayalam and Telugu. The recognition is performed in a bottom-up fashion, starting with the strokes, and the ambiguities at each stage are preserved and transferred to the next stage for obtaining the most probable results at each stage. The system achieves a stroke level accuracy of 95.78% and 95.12% on Malayalam and Telugu data, respectively. The akshara level accuracy of the system is around 78% on a corpus of 60, 492 words from 367 writers.

2.6.3 Syntactic or Structural Approach

When recognition problems involve complex patterns, it is more appropriate to adopt a hierarchical perspective where a pattern is viewed as being composed of simple sub patterns [80]. These simple sub patterns are built from yet simpler sub patterns. The term primitive is used to denote simplest and elementary sub pattern to be recognized and the given complex pattern is represented in terms of the interrelationships between these primitives. In syntactic pattern recognition, a formal analogy is drawn between the structure of patterns and the syntax of a language. The patterns are viewed as sentences belonging to a language, primitives are viewed as the alphabet of the language, and the sentences are generated according to a grammar. With the help of a small number of primitives and grammatical rules, a large collection of complex patterns can be described. The grammar for each pattern class must be inferred from the available training samples. In addition to classification, structural pattern recognition approach also provides a description of how the given pattern is constructed from the primitives. This paradigm has been used in situations where the patterns have a definite structure which can be captured in terms of a set of rules, such as waveforms, textured images, and shape analysis of contours [80]. Hammed et al. [89] proposed an efficient structural approach for recognizing on-line handwritten digits. After reading the digit from the user, the two dimensional coordinates of the pixels representing the digit are used for calculating and normalizing slope values of these coordinates. Successive slope values are then used to record the change of direction which used to estimate the slope. Based on the changing of signs of the slope values, the primitives are identified and extracted. These primitives represent a specific string which is a production of a certain grammar. Each digit can be described by a specific string. In order to identify the digit, the grammar is determined to which the string belongs. A Finite Transition Network containing the grammars of the digits is used for the matching of the string of the primitives with the corresponding digit to identify the digit. The proposed method is tested on a sample of 3000 digits written by 100 different persons, where each person wrote the 10 digits three times each. The method achieved accuracy of about 95% on the sample test. Chan et al. [90] proposed a robust structural approach for recognizing on-line handwriting which aims at achieving fairly high accuracy, reasonable speed and sufficient tolerance to variations. Simultaneously, it maintains a high degree of extensibility and reusability. The recognition rates obtained are 98.60% for digits, 98.49% for uppercase letters, 97.44% for lowercase letters, and 97.40% for the combined set. This work is an effective and efficient on-line character recognition module which is used as part of a pen-based mathematical equation editor developed by a syntactical pattern recognition approach. Kuroda et al. [91] proposed a syntactic pattern recognition based method for the recognition of online handwritten Chinese characters. For feature extraction Kohonen's self-organizing feature map was used, with an intention to get optimal sets of prototypical waveforms of peaks from sample data automatically. The strings of symbols are converted into matrices expressing features of the successors, and are analyzed by simple calculations between matrices. Additionally, in order to symbolize and analyze efficiently and accurately in a large scale, a hierarchical approach for the proposed method was employed. Using free writing characters, recognition rates of 99.49% for training patterns and 94.34% for test patterns were obtained. Joshi et al [13] describe a system for the automatic recognition of isolated handwritten Devanagari characters. Due to the large number of characters and resulting demands on data acquisition, a structural recognition technique was used to reduce some characters to others. The residual characters are then classified using the subspace method. Finally the results of structural recognition and feature-based matching are mapped to give final output. The proposed system is evaluated for the writer dependent scenario. Abdelwaheb et al. [92] proposed a system for the interpretation of online handwritten mathematical formulas based on a syntactic parser. This system is able to recognize a large class of mathematical formulas written on a graphic tablet. It starts the parsing by localization of the principal operator in the formula and attempts to partition it into sub expressions which are similarly analyzed by looking for a starting character. Garain et al. [93] introduced a method for automatic recognition of online handwritten mathematical expressions. Symbol recognition and structural analysis are the two major stages involved in the proposed technique. Two different classifiers have been combined to achieve high accuracy in recognition of symbols. To identify the spatial relationships among symbols, several online and offline features are used in the structural analysis phase. They also designed a context-free grammar to convert the input expressions into their corresponding $\TeX$ strings, converting subsequently them into MathML format. Assabie et al. [94] introduced an online handwriting recognition system for Ethiopic script based on the structural and syntactical analysis of the strokes forming characters. The complex structures of characters are represented by the spatiotemporal relationships of primitives. A special tree structure is used to model spatiotemporal relationships of the strokes. The tree generates a unique set of primitive stroke sequences for each character, and for recognition each stroke sequence is matched against a stored knowledge base. Characters are also classified based on their structural similarity to select a reasonable set of characters for unknown input, which improves recognition and processing time.

2.6.4 Neural Networks

Neural networks can be regarded as parallel computing systems consisting of an extremely large number of simple processors with many interconnections [79, 80]. Neural network models attempt to use learning, generalization, adaptability, fault tolerance and distributed representation and computation in a network of weighted directed graphs in which the nodes are artificial neurons and weighted directed edges are connections between neuron outputs and neuron inputs. The ability to learn complex nonlinear input-output relationships, use of sequential training procedures, and adaptability to the data are the main characteristics of neural networks. Feed-forward network is the most commonly used family of neural networks for pattern classification tasks, which includes multilayer perceptron and Radial-Basis Function (RBF) networks. These networks are organized into layers having unidirectional connections between the layers. Self-Organizing Map (SOM), or Kohonen-Network is another popular network which is mainly used for data clustering and feature mapping. Guyon et al. [38] described a system for online handwriting recognition based on neural network. The system can recognize digits and uppercase letters hand printed on a touch terminal. The constituent sequence of coordinate points of the character is subjected to very simple preprocessing, and then classified by a trainable neural network. The network was trained on a set of 12,000 digits and uppercase letters, from approximately 250 different writers, and tested on 2500 such characters from other writers. Classification accuracy exceeded 96% on the test examples. Bellegarda et al. [95] performed online handwritten character recognition using multilayer feed forward neural networks. A combination of parallel networks is developed to overcome commonly encountered difficulties such as slow training process and requirement for a large amount of training data. The recognition system has been evaluated, on tasks which involve discrimination between similarly shaped characters and recognition of discretely written upper-case characters. Verma et al. [35] introduced a technique for online handwriting recognition based on Artificial Neural Networks (ANN). The technique creates a single global feature vector by combining directional, structural, and zoning information. The technique is independent to character size and therefore it can extract features from the raw data without resizing. Experiments were conducted on UNIPEN benchmark database using the proposed feature vector and a Neural Network based classifier. The recognition rates obtained were 98.2% for digits, 91.2% for uppercase and 91.4% for lowercase. Sampath et al. [42] proposed a method for online handwritten Malayalam character recognition based on neural network. The method aims at training a simple neural network with three layers using back propagation algorithm. Freeman codes are used to represent each character as feature vector. These feature vectors act as inputs to the network during the training and testing phases of the neural network. The output is the character expressed in the Unicode format. Kwaik et al. [96] introduced an online handwritten recognition system for isolated Arabic characters based on Feed forward back propagation neural networks process. Due to the low computation overhead during training and recall process, feed forward back propagation neural networks approach is employed as classifiers. The system achieves an recognition rate 95.7% from untrained writers and 99.1% for trained writers. Kubatur et al. [97] proposed a neural network based framework to classify online Devanagari characters into one of 46 characters in the alphabet set. The system makes use of the Discrete Cosine Transform of the temporal sequence of the character points as features. The testing has been carried on 2760 characters, and recognition rates of up to 97.2% are achieved.