Chapter 3

Preprocessing Techniques for On-Line Handwriting

3.1 Introduction

Most of the classification techniques assume that the data is given in a predetermined form, which satisfy certain requirements as to quality, size, invariance, etc. However, these characteristics are commonly not satisfied by on-line handwritten data, as we can see in Fig. 3.1. The low quality of the data is due basically to the combination if three facts. One is the addition of noise during digitalization, which is generally generated by a badly configured digital tablet. The other is the irregularity generated by inexperienced users having an erratic handwriting. The last are variations in handwriting styles.

To overcome these problems, we use preprocessing, which involves the substitution, removal, reordering, and/or extraction of the data. Preprocessing eliminates noise, normalizes handwriting, and reduces the amount of redundant information, in order to fix the variations of handwriting and facilitate encoding of raw data into feature vectors.

Throughout this chapter we will represent a stroke s as an ordered sequence of points $\{p_i\}_{i=1}^{n}$, where $p_i = (x_i, y_i)$. The points p_1 and p_n correspond to the first and last touches of the stylus. New data obtained by preprocessing is labeled with an asterisk.

This chapter is organized as follows. Section 3.2 refers to preprocessing methods for noise and data reduction. Section 3.3 explains the methods for handwriting normalization we used. Section 3.4 refers to a method for artificial symbol generation and

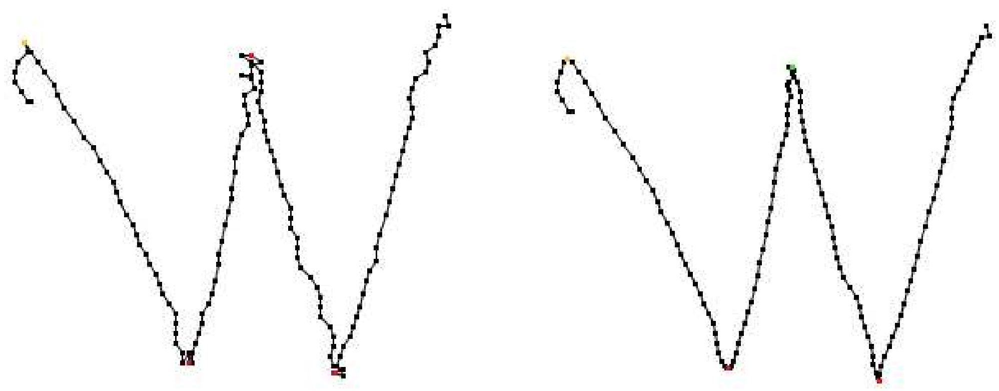
Figure 3.1: *Left: original data. Right: data after smoothing. We used the coefficients* $\alpha_{-1} = 1/4$, $\alpha_0 = 1/2$, *and* $\alpha_1 = 1/4$.

Sect. 3.5 describes the symbol's features used to construct the input of the classifiers.

3.2 Noise and Data Reduction

3.2.1 Smoothing

Smoothing is one of simplest approaches for data filtering. See Fig. 3.1. As most preprocessing methods, it consists of substituting the coordinates of the original point. It is done using a weighted sum of the neighboring points:

$$p_i^* = \sum_{k=-n}^{n} \alpha_k p_{i+k}, \tag{3.1}$$

where

$$\sum_{k=-n}^{n} \alpha_k = 1.$$

The most commonly-used coefficients are $\alpha_k = 1/(2n+1)$ and coefficients generated by a discrete approximation of a Gaussian distribution, as shown in Fig. 3.2. By default, we use three coefficients with values $\alpha_{-1} = 1/4$, $\alpha_0 = 1/2$, and $\alpha_1 = 1/4$.

3.2.2 Point Clustering

This kind of filtering also involves averaging with neighboring points. In this case, we consider the neighboring points of p_i that belong to a predetermined vicinity $V_r(p_i)$ with a radius $r > 0$, where

$$V_r(p_i) = \{p \in s : \|p_i - p\| <= r\}.$$

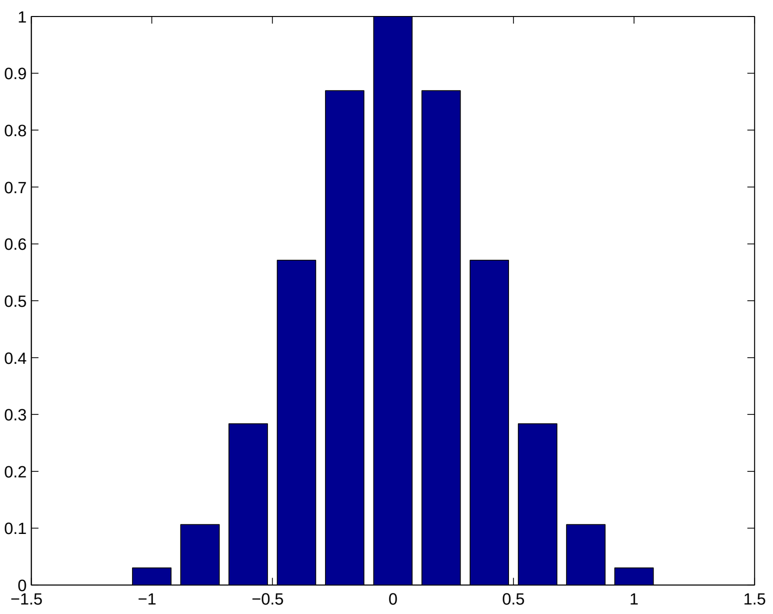

Figure 3.2: *Coefficient obtained by a discrete approximation of a Gaussian distribution. In this case, eleven coefficients are used in the filter.*

The original point is substituted by

$$p_i^* = \frac{1}{\#(V_r(p_i))} \sum_{p \in V_r(p_i)} p,$$

where # means the number of points in the neighborhood. In this way we obtain the new stroke $s^* = \{p_{i_k}^*\}_{k=1}^m$, where

$$i_{k+1} = \min\{i_k < i \leq n : p_i \notin V_r(p_{i_k})\},\ k = 2, \ldots, m-1,\ i_1 = 1. \tag{3.2}$$

This filter smooths the stroke and removes repeated points. This method also helps to remove some features which are smaller than the given radius. Figure 3.3 shows that the little hook on the left symbol is removed, as well as the points clustering in a stroke's corners. In some cases, points of the filtered strokes tend to have a regular separation between them. This occurs when the distance between consecutive points in the stroke is smaller than the neighborhood radius. The default value we used for the radius is $L/80$, where L represents the length of the stroke.

3.2.3 Dehooking

Hooks are very common artifacts found at the ends of the strokes, see Fig. 3.4. They are generated during fast writing, when pen-down and pen-up events are generated

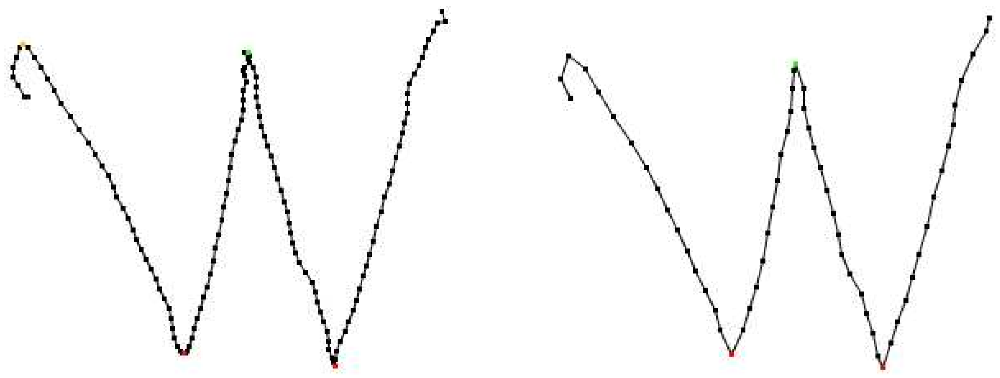

Figure 3.3: *Left: original data. Right: data after clustering.*

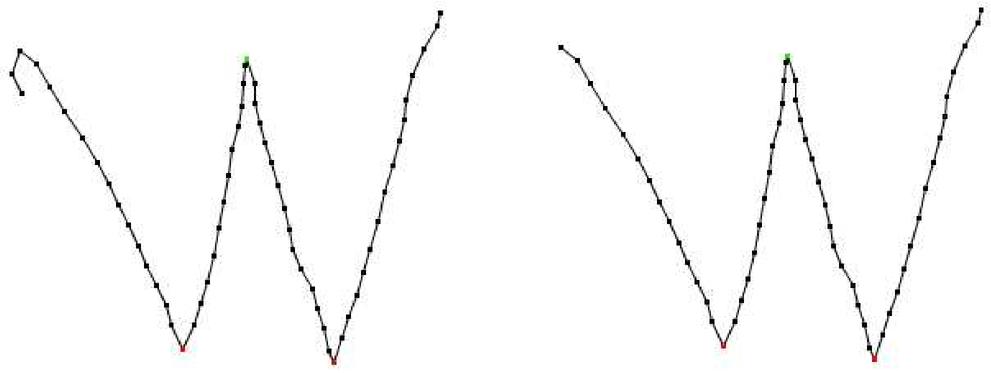

Figure 3.4: *Left: original data. Right: data after dehooking.*

with a delay, such that the events do not match with the real touch and lifting of the stylus. The way hooks are detected in strokes consists of locating abrupt changes of the *turning angle*. Given a point p_i in the stroke, the turning angle θ_i is formed by the consecutive line segments $\overline{p_{i-1}p_i}$ and $\overline{p_i p_{i+1}}$, see Fig. 3.5.

To eliminate the parts of a stroke which constitute a hook, two conditions must be met. The first one is that the turning angle satisfies $\theta_i > \theta$, where θ is a given threshold. The second condition is that the point p_i satisfies some of the the arc length conditions

$$\sum_{k=1}^{i-1} \|p_{k+1} - p_k\| \quad < \quad \alpha L, \tag{3.3}$$

$$\sum_{k=i}^{n-1} \|p_{k+1} - p_k\| \quad < \quad \alpha L, \tag{3.4}$$

where alpha is a real number $0 < \alpha < 1$ and L is the stroke's length. Conditions (3.3)-

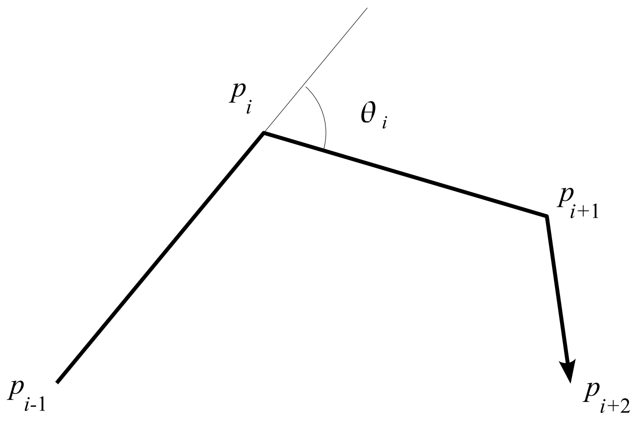

Figure 3.5: *Turning angle.*

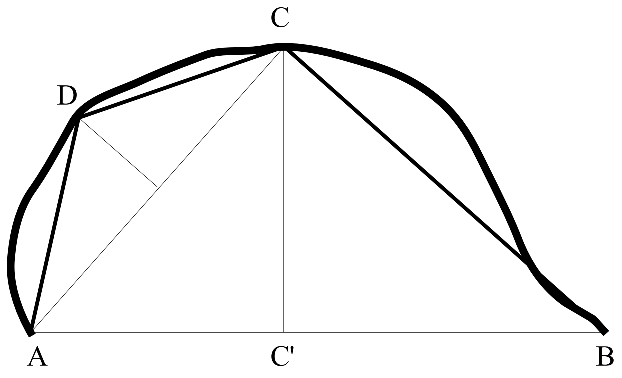

Figure 3.6: *Two recursion steps of the method for polygonal approximation.*

(3.4) correspond the hook analysis at initial and end points respectively. The default values we used are $\theta = 85$ degrees and $\alpha = 0.12$.

3.2.4 Polygonal Approximation

To approximate strokes using a polygonal line with a fixed number of points, we proceeded in the following way. We start by considering the segment $\overline{AB}$ formed by the first and the last point of the stroke, see Fig. 3.6. This segment is the basis of the triangle formed with point C of the stroke. Among all the points of the stroke, we select the one which reaches the maximal height $\overline{CC'}$ of the triangle ABC. The first approximation of the stroke is formed by the points $\{A, C, B\}$. We repeat the procedure to the two the sub-strokes $\{p_1, \ldots, p_k\}$ and $\{p_k, \ldots, p_n\}$, where $p_1 = A$, $p_k = C$, and $p_n = B$. The procedure finishes when a desired number of points is reached. A variation of the method consists of taking the minimum area of the

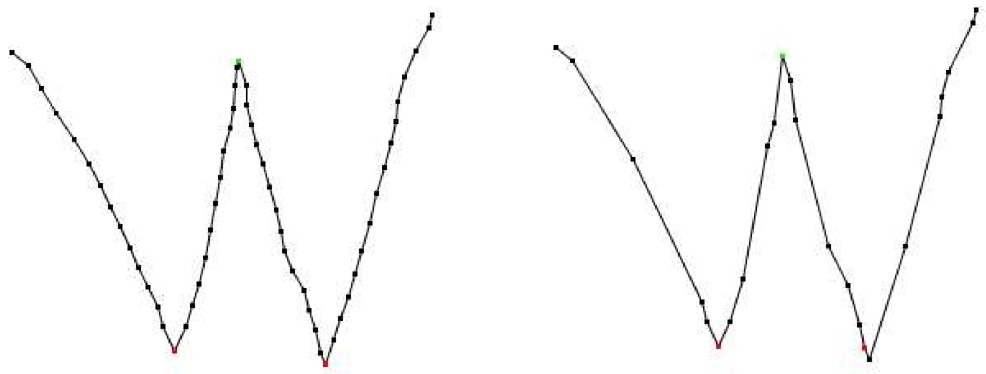

Figure 3.7: *Left: original data. Right: result after applying polygonal approximation.*

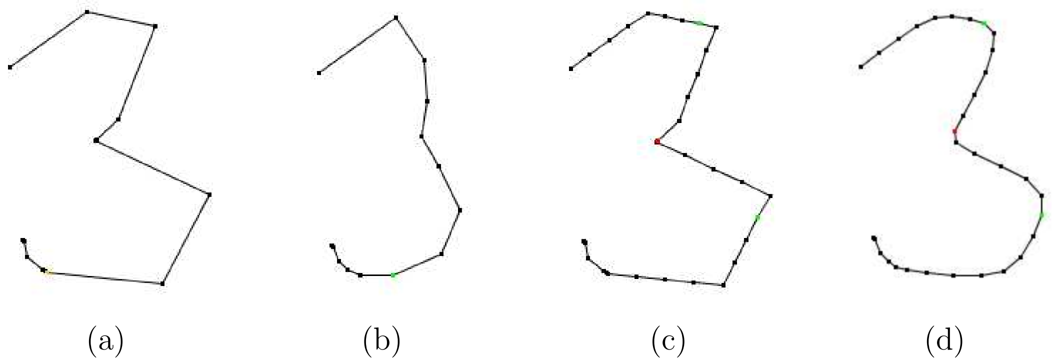

Figure 3.8: *(a) Original symbol. (b) Original symbol after smoothing. (c) Original symbol after adding extra points. (d) The symbol in (c) after smoothing.*

triangle instead of its height.

Sometimes, we need the opposite procedure, namely addition of points, because the number of points is so low that some undesirable results can be obtained when applying preprocessing methods, as shown if Fig. 3.8. In this case, the sampling rate of the handwriting device is decreased and only few points are stored. As we can see, smoothing degrades the data and an incorrect label could be assigned to the symbol. To overcome this problem we add points as follows. A recently added point corresponds to the middle point of the longest segment in the stroke. This procedure is repeated until the desired number of points is reached.

3.2.5 Arc Length Resampling

In this preprocessing step, we obtain new samples of points that are regularly spaced with respect to the *arc length*. The method which is used to resample new points is

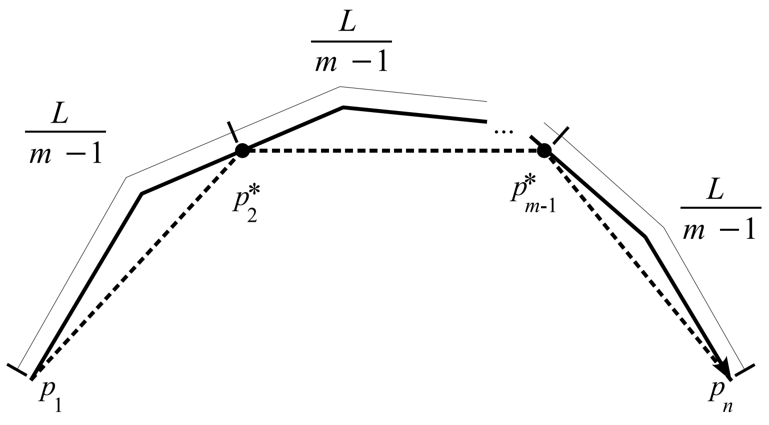

Figure 3.9: *Result of arc length resampling. The solid line represents the original stroke while the dashed one represents the new stroke obtained by linear interpolation.*

a simple linear interpolation, see Fig. 3.9. Suppose that the stroke $s = \{p_i\}_{i=1}^n$ has length L. The new stroke s^* is initialized with the point p_1. i.e. $s^* = \{p_1\}$. If $L \neq 0$ and $n > 1$, we can find an index $1 \leq k \leq n$ such that

$$\sum_{i=1}^{k-1} \|p_{i+1} - p_i\| \leq \frac{L}{m-1} < \sum_{i=1}^{k} \|p_{i+1} - p_i\|, \tag{3.5}$$

where m is the desired number of points and L the length of the stroke. We find the point p_2^* by interpolating linearly in the segment $\overline{p_k p_{k+1}}$ such that

$$\sum_{i=1}^{k-1} \|p_{i+1} - p_i\| + \|p_2^* - p_k\| = \frac{L}{n-1}. \tag{3.6}$$

We then reinitialize $s^* = \{p_1, p_2^*\}$. We repeat the procedure with the stroke $\{p_1^*, p_{k+1}, \ldots, p_n\}$ to construct the rest of the points $p_3^*, \ldots, p_{m-1}^*, p_m^* = p_n$ that satisfy the arc length condition (3.6). See Fig.3.10.

3.3 Normalization

3.3.1 Stroke Grouping

To group strokes into single symbols, we proceed as follows. Two strokes belong to the same group if their mathematical dilation with a circle of radius αr intersects some of the strokes of the symbol, where r is the current stroke thickness and $\alpha > 1$, see Fig. 5.7. The result of dilating a stroke with the circle is equivalent to increasing the stroke's thickness to $(1 + \alpha)r$.

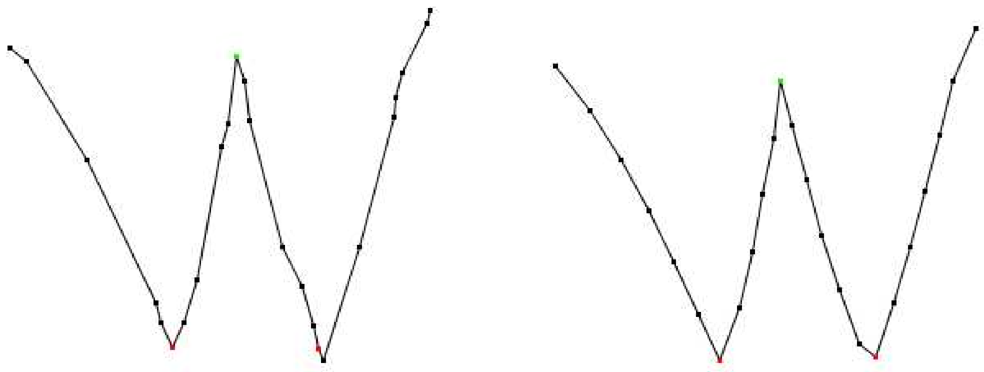

Figure 3.10: *Left: original symbol. Right: symbol processed by arc length resampling.*

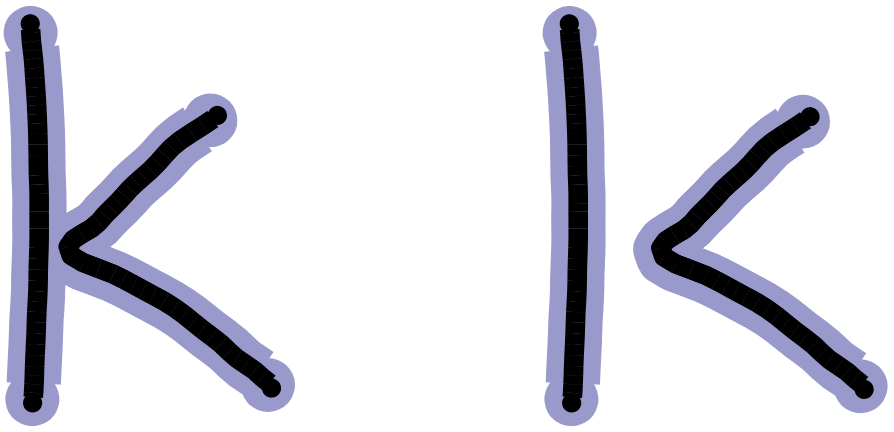

Figure 3.11: *Groping of strokes. Left: dilation of strokes intersects. Right: dilation of strokes does not intersect.*

To decide wether two strokes form a single one, we use the following criterion. If the distance of the end points in the strokes is lower than αr, it is assumed that the last stroke is a continuation of the previous stroke. In this case, both strokes are concatenated, instead of being grouped together as one symbol. The default value we use for both criteria is $\alpha = 1.5$.

3.3.2 Stroke's Direction and Order

We experienced that left-handed users write some symbols and letters in the opposite direction of right-handed ones. For this reason we use the following method to normalize the direction of strokes [61]. First, we classify each stroke as *closed*, *horizontal*, *vertical*, or *diagonal*, using the ratios

$$R_x = |x_n - x_1|/D \text{ and } R_y = |y_n - y_1|/D, \tag{3.7}$$

where D is the length of the diagonal of the symbol's bounding box. Thus, we select a threshold $\delta \in [0, 1]$ and we say that stroke s is

- **closed** if
$$R_x < \delta \text{ and } R_y < \delta, \tag{3.8}$$
- **horizontal** if
$$R_x \geq \delta \text{ and } R_y < \delta, \tag{3.9}$$
- **vertical** if
$$R_x < \delta \text{ and } R_y \geq \delta, \tag{3.10}$$
- **diagonal** if
$$R_x \geq \delta \text{ and } R_y \geq \delta. \tag{3.11}$$

The direction of horizontal strokes are changed if $x_l < x_f$. Vertical and diagonal strokes are normalized if $y_l < y_f$. This is done by changing the direction. Given a stroke s, a new stroke obtained by changing the direction takes the form

$$s^* = \{p_{n+1-i}\}_{i=1}^{n}. \tag{3.12}$$

Once the direction of strokes is normalized, they are ordered respect to the angle formed by two segments. One corresponds to the upper segment of the symbol's bounding box and the other is formed by the upper left corner of the bounding box and the last point of the stroke. See Fig. 3.12.

3.3.3 Stroke Reduction

Sometimes, it is desirable that symbols have a fixed number N of strokes. When the number of strokes in a symbol exceeds this number, we concatenate strokes as follows. Assuming that the strokes in the symbol are ordered as described in the last section, we keep the first $N - 1$ strokes and simply concatenate the rest to form the N-th stroke.

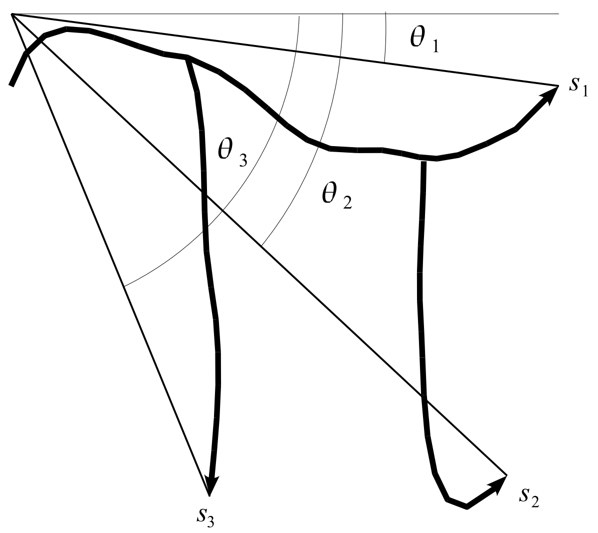

Figure 3.12: *Stroke reordering.*

3.3.4 Size Normalization

Symbols are scaled and translated such that their bounding box fits the square $[-1, 1] \times [-1, 1]$ and the center of their bounding box and the center of the square coincide. Note that the ratio between height and width is kept constant.

3.4 Artificial Symbol Generation

Sometimes, there are not enough data to train the classifiers. We overcome this problem by generating symbols *artificially.* Schwenk and Milgram [79] define a basic set of transformations on symbols by representing a stroke s as a vector formed by coordinate points

$$s = (x_1, y_1, \ldots, x_n, y_n)^T, \tag{3.13}$$

and transform it linearly into $t(s, \alpha)$ using the expression:

$$t(s, \alpha) = s + \alpha t. \tag{3.14}$$

The vector t determines the transformation, and α is a real value. Seven basic transformations are defined by specifying the values of the vector $t = (x_1^t, y_1^t, \ldots, x_n^t, y_n^t)$:

1. **Translation x-Axis:**
$$x_i^t = 1 \text{ and } y_i^t = 0. \tag{3.15}$$
2. **Translation y-Axis:**
$$x_i^t = 0 \text{ and } y_i^t = 1. \tag{3.16}$$
3. **Scale:**
$$x_i^t = x_i \text{ and } y_i^t = y_i. \tag{3.17}$$
4. **Axis deformation:**
$$x_i^t = -x_i \text{ and } y_i^t = y_i. \tag{3.18}$$
5. **Rotation:**
$$x_i^t = -y_i \text{ and } y_i^t = x_i. \tag{3.19}$$
6. **Diagonal deformation:**
$$x_i^t = -y_i \text{ and } y_i^t = x_i. \tag{3.20}$$
7. **Slope-x:**
$$x_i^t = \Delta x_i \text{ and } y_i^t = 0. \tag{3.21}$$
8. **Slope-y:**
$$x_i^t = 0 \text{ and } y_i^t = \Delta y_i. \tag{3.22}$$

For the slope transformation, we use the following formula to calculate the discrete derivative:
$$\Delta u_i = \begin{cases} u_2 - u_1, & i = 1, \\ \frac{1}{2}(u_{i+1} - u_{i-1}), & 1 < i < n, \\ u_n - u_{n-1}, & i = n. \end{cases} \tag{3.23}$$
This kind of transformation corresponds to a change in the local curvature, see Fig. 3.13.

The most general transformation takes the form:
$$f(s, \alpha) = s + \sum_{k=1}^{8} \alpha_k t_k, \tag{3.24}$$
where the vector t_k, $k = 1, \ldots, 8$, corresponds to the transformations listed before.

To generate new symbols, we randomly select different values for the parameter values α_k. We can see from (3.15) and (3.19) that using the values of $\alpha = -1$ and

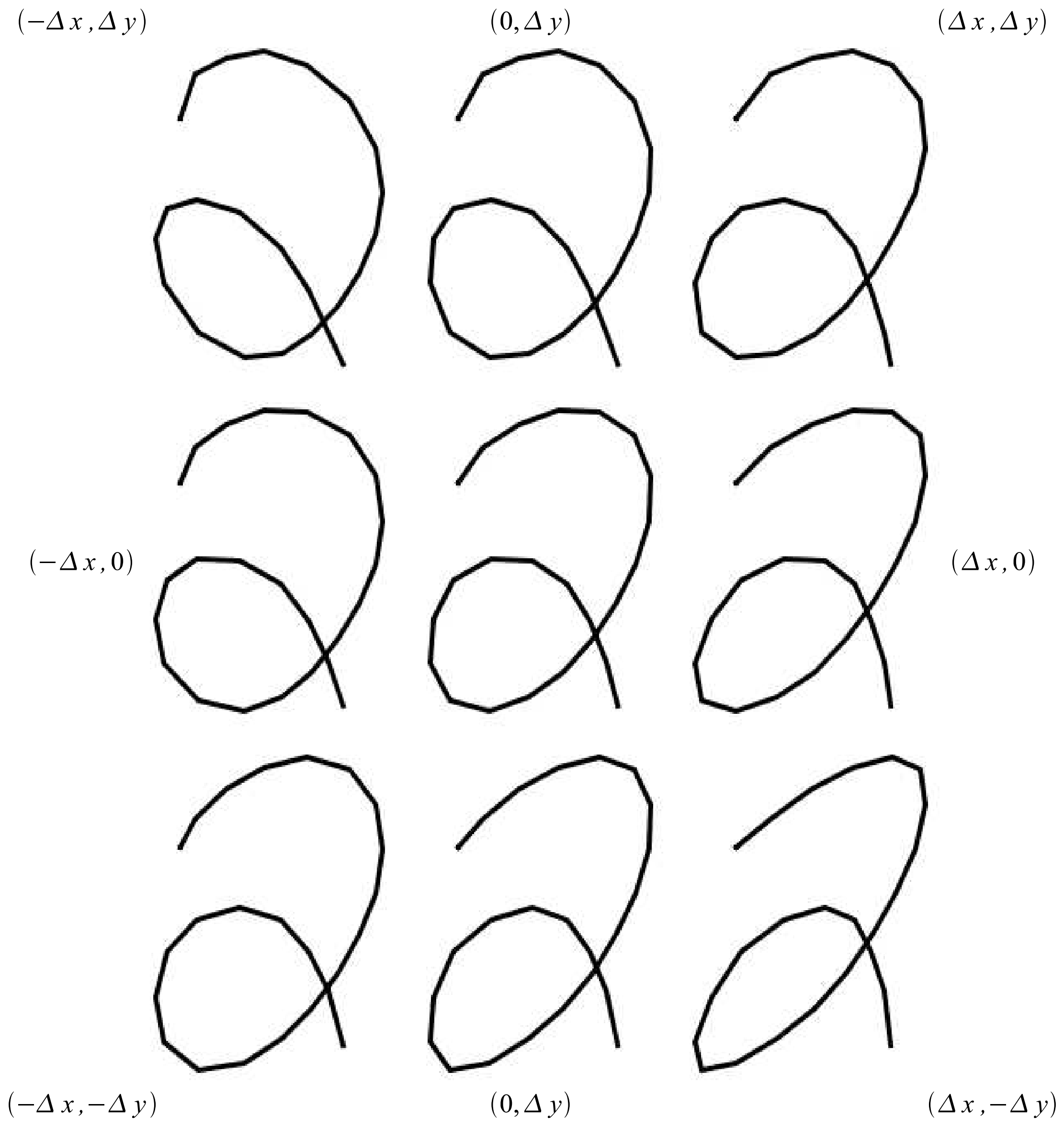

Figure 3.13: *Slope transformation results. The original symbol is shown at the center.*

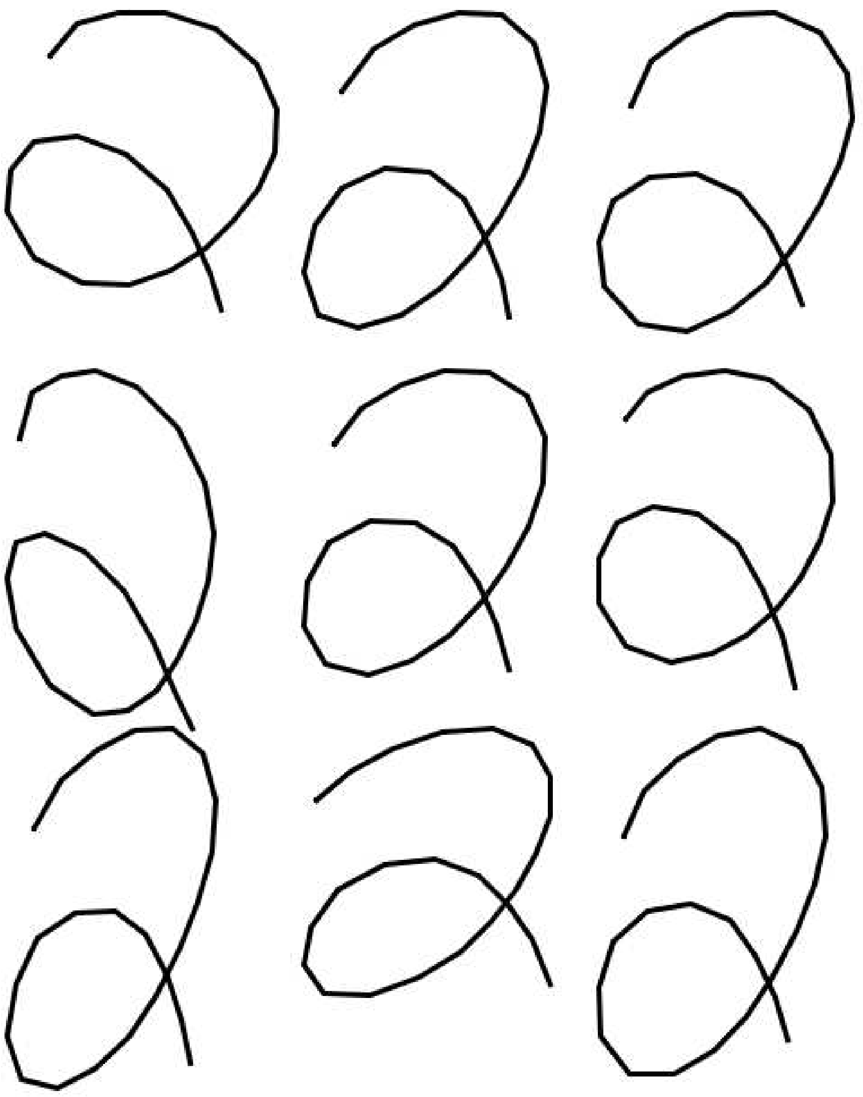

Figure 3.14: *Random transformation results. The original symbol is shown at the center.*

$\alpha = -2$ deforms the stroke in a single point and rotates it by 180 degrees respectively. To avoid this extreme deformation of symbols, we select "small" values for α_k: we take α randomly from the interval $(-0.15, 0.15)$, except for the slope transformation. In that case we use the values $(-0.3, 0.3)$. Actually, the selection of the limits for these intervals depends strongly on the number of points the stroke contains. Figures 3.13-3.14 show transformations of the symbol '2 using twenty-four points in each stroke.

3.5 Feature Extraction

Feature extraction means deriving measures and characteristics from the raw data that are useful in making predictions. Commonly feature extraction methods are based on invariance, reconstruction, and expected distortions. We studied several local and global features that are normally used by authors as mentioned in Chapt. 2. Among those features, we found the following to yield the best results.

3.5.1 Local Features

Given a stroke $s = (p_1, \ldots, p_n)$ of length ℓ, we use the following *local features* to construct the input vector for the classifier:

Point Coordinates

The values (x_i, y_i) of p_i are the main attributes and are the base to derive others from.

Turning Angle

We use the values $\sin(\theta_i)$ and $\cos(\theta_i)$ of the turning angle θ_i at the point p_i. The are calculated by the expressions

$$\cos(\theta_i) = \frac{x_{i+1} - x_i}{d(p_i, p_{i+1})} \quad \text{and} \quad \sin(\theta_i) = \frac{y_{i+1} - y_i}{d(p_i, p_{i+1})}, \tag{3.25}$$

where $i = 1, \ldots, n-1$. These features describe the direction of the stroke between consecutive points.

Turning Angle Difference

The values

$$\sin(\theta_{i+1} - \theta_i) \quad \text{and} \quad \cos(\theta_{i+1} - \theta_i), \tag{3.26}$$

represent the turning angle difference for $i = 1, \ldots, n-2$. They describe the curvature of the stroke at the i-th point.

Length Position

The length position of p_i is defined as

$$L_i = \sum_{k=1}^{i-1} d(p_k, p_{k+1})/L, \tag{3.27}$$

where L represents the length of the stroke. This measurement is useful to locate points, irrelevant of the scale, in strokes having no equidistant points.

3.5.2 Global Features

The following global features of s are also used:

Center of Gravity

The center of gravity

$$\left(\sum_{i=1}^{n} x_i/n, \sum_{i=1}^{n} y_i/n\right) \tag{3.28}$$

is used to locate strokes for symbols formed by several strokes. This measurement helps to distinguish symbols formed by strokes with similar structures.

Length

The length of a stroke is a useful measure to distinguish between long and short strokes.

Relative Length

The relative length is defined as

$$L_r = d/L, \tag{3.29}$$

where d is the distance between the first and last points of s. This measure describes how “closed” a stroke is. The value $L_r \approx 0$ corresponds to a “closed” stroke, while $L_r = 1$ to a stroke which is actually a line segment.

Accumulated Angle

The accumulated angle

$$\theta_a = \frac{1}{2\pi} \sum_{i=1}^{n} \theta_i, \tag{3.30}$$

where θ_i is the turning angle at point i; also describes how “closed” a stroke is in the sense of angles.

Quadratic Error

This feature represents the sum

$$\ell = \frac{1}{n} \sum_{i=1}^{n} d_i^2, \tag{3.31}$$

where d_i is the distance between the point p_i and the line segment formed by the end points of the stroke. This measure describes how "similar" a stroke is to a line segment.