

# NEW PREPROCESSING TECHNIQUES FOR HANDWRITTEN WORD RECOGNITION

M. Blumenstein, C. K. Cheng and X. Y. Liu  
School of Information Technology, Griffith University  
PMB 50, GCMC Bundall 9726  
Queensland, Australia  
E-mail: [m.blumenstein@mailbox.gu.edu.au](mailto:m.blumenstein@mailbox.gu.edu.au)

## Abstract

The research described in this paper focuses on the presentation of two novel preprocessing techniques for the task of off-line handwritten word recognition. A technique for the identification of straight and skewed underline noise is described along with a novel algorithm for detecting skew in handwritten words. The latter identifies skew by detecting the center of mass in each half of a word image. By hypothesizing a line between the two centres and by measuring the angle it makes with the x-axis, an angle for skew may be estimated. The algorithms are tested on the CEDAR benchmark database of handwritten cursive words. Results above 96% are reported for skew detection and underline removal.

**Key Words:** Handwritten Word Recognition, Preprocessing, Skew Detection, Underline Removal

## 1. Introduction

When considering the difficult problem of automated recognition of off-line handwritten words, many phases need to be implemented in order to achieve high accuracy. One of the most important steps required for handwritten word recognition is that of preprocessing. The task of preprocessing relates to the removal of noise and variation in handwritten word patterns [1]. Preprocessing may itself be broken down into smaller tasks such as noise removal, slant estimation and correction, skew detection, resizing etc. By preprocessing each handwritten word, the task of subsequent recognition is simplified.

In many real-world applications, user input through the medium of handwriting on a static surface is unconstrained. Handwriting varies in slant and skew as well as the amount of noise and ornamentation that may be introduced through individual writing style and context [2]. In previous research on a benchmark database of handwritten postal words [3], it was noted that a small percentage of word images were skewed and contained underlines amongst other variations. Previous research has indicated to us that the effect of the aberrations mentioned above impacted highly on the success of our recognition system.

In the literature, most techniques for the preprocessing of handwritten words are described as part of an overall system for handwriting recognition [2],[4],[5],[6]. In the references cited, the individual techniques for noise removal and skew detection are not evaluated outside of their systems. In [2], Senior and Robinson describe a skew detection technique whereby minima in the lower contour of the word image are first located and a line of best fit is drawn through these points. As mentioned earlier, the technique in the above-mentioned research is not evaluated separately from their proposed system. Also, their preprocessor does not take into account the possibility of skewed underlines in the handwriting tested.

Few researchers have evaluated their individual preprocessing components, an example is Dimauro *et al's* underline removal technique [7]. The authors use mathematical morphology for removing underlines in handwritten words. Although their system performs well, it does not seem to take into account the possibility of skewed handwriting.

In this paper we describe new techniques for the accurate removal of underlines and detection of skew. The underline removal technique targets simple straight underlines as well as the more difficult skewed underlines inherent in some words. Underlines are detected through the analysis of horizontal black pixel runs, the word's vertical histogram, as well as word skew. Underline removal is accurately executed through first measuring the stroke thickness of each word and obtaining an average value that can be used as a threshold for removing underline noise.

Skew detection is achieved through elimination of word ascender and descender information and using the center of mass of relevant components of each word.

The techniques mentioned above are tested on the CEDAR benchmark database. High accuracy for each technique is reported and discussed.

The remainder of this paper is divided into 4 sections. Section 2 describes the proposed preprocessing techniques in detail, Section 3 details the results obtained, Section 4 presents a discussion of the results and finally conclusions and future research are presented in Section 5.

## 2. Proposed Preprocessing Techniques

In the sections that follow, the algorithms described have been tested on binary word images. The black pixels in the binary images represent the handwriting (foreground pixels) whilst white pixels are used to denote the background. From this point on, the terms "black pixels" and "foreground pixels" are used interchangeably.

### 2.1 Removal of Underlines

In this research, three categories of underlines have been targeted for removal: 1) Straight Underlines, 2) Underlines located in the lower half of the word image and 3) Underlines that match the slope of the word. Three related algorithms have been developed to detect each category of underline. The first two types of underlines may be detected prior to word skew detection and shall be described in this section. The final category is discussed in Section 2.2.

#### 2.1.1 Straight Underline Removal

The accurate detection of upper and lower baselines as well as word skew can be adversely effected by abnormal distributions of foreground pixels in handwritten words. The existence of straight underlines contributes to the above-mentioned noise. Hence the first component of our preprocessing technique aims at detecting these underlines.

##### 2.1.1.1 Measurement of Average Stroke Thickness

Prior to underline removal, it is necessary to determine the approximate thickness of strokes in each word. This will ensure that if a straight line is located, strokes present in letters such as 'y' and 'g' that overlap the underline will not be removed in the process. It is assumed that stroke thickness will be similar to the thickness of the underlines present in a word.

The algorithm for measurement of average stroke thickness is described below:

1. Starting at the top of each column in the word image and proceeding to the bottom, sum the number of foreground pixels contained in the last continuous run.
2. Sum all the above-mentioned runs and calculate the average. This is the initial value for average stroke thickness.
3. Return to Step 1 so that outlier pixel runs caused by letters such as 'l', 't', 'k' etc. may be removed.
4. Only pixel runs that are smaller than the initial value for average stroke width are summed and averaged. This will generate a stroke thickness closer to the actual value.

#### 2.1.1.2 Straight Line Removal

The algorithm for removing straight lines is described below:

1. For each row of the word image, examine each run of continuous foreground pixels.
2. Note the length of each run.
3. IF a particular run is greater than half of the actual word length THEN  
    A straight underline has been found
4. Store the x-coordinates of the start and end column of the line
5. The underline's stroke width is examined by measuring the length of vertical runs of foreground pixels in each column between the two stored x-coordinates.
6. IF the length of a continuous run of foreground pixels in the above-mentioned area is smaller than or equal to the average stroke width THEN  
    All pixels in that run are converted to background pixels.  
ELSE  
    The foreground pixels remain unchanged

#### 2.1.2 Removal of Underlines in the Lower Section of the Word

Prior to the detection and removal of underlines in the lower section of the word, it is first necessary to define the location of the "lower section" and to obtain preliminary values for the upper and lower baseline of the word. The process of baseline detection is defined and described in the next section.

##### 2.1.2.1 Baseline Detection

Some strokes in a word image may extend above or below the middle zone or main body of a handwriting sample. Such letter components are called ascenders and descenders respectively [2]. Examples of letters that contain such strokes are: 'f', 'j', 'g' etc. Hence the middle zone of a word image that does not contain ascenders and descenders is bounded by an upper and lower baseline. In the context of our proposed techniques, baseline detection is required for two reasons. Firstly, preliminary baseline information assists in detecting underlines in a handwritten word image. Secondly, ascenders and descenders have been found to influence the success of our skew detection technique, and hence it is necessary to incorporate baseline estimation to facilitate the removal of ascender and descender information.

The technique used in this research is a modified version of that described in [4]. A brief overview is described below:

1. The height of the word is measured and divided into four equal components

2. The row containing the largest run of black pixels in the word image is located. This is named the "Peak Line".
3. The average number of foreground pixels in each row is calculated
4. A vertical histogram of foreground pixels in the image is analysed
5. Determining the Upper baseline:
  - a. The row containing the minimum number of foreground pixels prior to the Peak Line is located and marked
  - b. Commencing from the marked row, find the first row containing a number of foreground pixels greater than or equal to the average number. This line is marked as the Upper baseline (upperBaseline).
6. Determining the Lower baseline:
  - a. The row containing the minimum number of foreground pixels after the Peak Line is located and its y-coordinate is stored (minPixY)
  - b. Commencing upwards from minPixY, find the first row containing a number of foreground pixels greater than or equal to the average number of foreground pixels (avgPixY).
  - c. IF (minPixY - avgPixY) is smaller than (avgPixY - upperBaseline) THEN  
                  lowerBaseline = minPixY  
          ELSE  
                  lowerBaseline = avgPixY

### 2.1.2.2 Underline Detection and Removal

Following preliminary baseline detection as described in Section 2.1.2.1, it is necessary to investigate the existence of a skewed underline in the lower section of the word. If an underline exists, it is removed as shown below:

#### Underline Detection:

1. The "lower section" of a word image is defined as beginning at the commencement of its lower quarter (Section 2.1.2.1, Step 1)
2. IF the Peak Line (Section 2.1.2.1, Step 2) exists in the lower quarter THEN  
          A skewed underline may be present.
3. To determine if an underline is present, foreground pixel runs in each column of the word between the upper and lower baseline are examined
4. IF a foreground pixel run is equal to the average stroke width of the word (Section 2.1.1.1) THEN
  - A potential underline component exists
  - Increment the length of the potential underline by 1
5. IF the length of the potential underline is greater than half the length of the word THEN  
          An underline has been found

#### Underline Removal:

1. The underline is removed by traversing the word from the lower baseline upwards and locating the first foreground pixel run in each column
2. IF a particular foreground pixel run is smaller than or equal to the average stroke thickness (Section 2.1.1.1) THEN  
                  Remove the pixel run

In the case where a skewed underline is found, it is necessary to recalculate the baseline values as the preliminary estimation was influenced by the existence of a false Peak Line (Section 2.1.2.1).

## 2.2 Skew Detection and Further Underline Removal

### 2.2.1 Skew Detection using the Centre of Mass

The skew detection technique described in this research is very simple and very fast. The algorithm is described below:

1. Following ascender and descender removal, the center of the word image is located and it is cut vertically into two equally sized components
2. The center of mass for each component is located using the formulae below:

$$XCentroid = \frac{\sum(x_{ij} * i)}{n}$$

$i = 0, \dots, NC-1$  where NC is the width of the component

$j = 0, \dots, NR-1$  where NR is the height of the component

$x = \{0,1\}$  Indicates the value of the current pixel being examined

$n =$  Number of foreground pixels

$$YCentroid = \frac{\sum(y_{ij} * j)}{n}$$

$i = 0, \dots, NC-1$  where NC is the width of the component

$j = 0, \dots, NR-1$  where NR is the height of the component

$y = \{0,1\}$  Indicates the value of the current pixel being examined

$n =$  Number of foreground pixels

Hence, the Centre of Mass may be represented by:

$$CoM = (XCentroid, YCentroid)$$

3. Once the x and y centroid coordinates for each component have been located it is possible to hypothesise a line that joins the two sets of coordinates (See Figure 1)
4. The slope of the word may be calculated using the following formula:

$$\text{Slope} = (y_2 - y_1) / (x_2 - x_1)$$

where,

$$\text{CoM}_{\text{RightComponent}} = (x_2, y_2)$$

$$\text{CoM}_{\text{LeftComponent}} = (x_1, y_1)$$

Angle of skew is finally found:

$$\theta = \tan^{-1}(\text{slope})$$

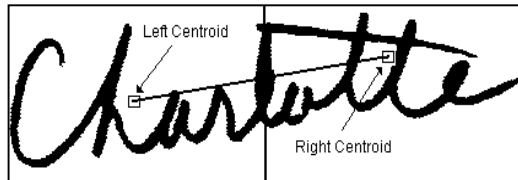


Figure 1. Centroid location and hypothesized line

Once the angle of skew has been detected, image rotation may be performed about the center of the image using simple geometric operations [9].

### 2.2.2 Removal of Underlines Matching the Slope of the Word

Following detection of word skew, any *straight* underlines or underlines existing in the bottom quarter of the word would have already been removed. However, in our research we have identified one final type of underline. In practice this final type matches the slope of the word image it resides in. The algorithm for eliminating this type of underline is described below:

1. Commencing at the left-most OR next available coordinate of the upper baseline, traversing downwards towards the lower baseline, examine each pixel and search for a foreground pixel
2. Once a foreground pixel is found, examine the next (adjacent) pixel using the slope found during skew detection
3. IF the next pixel is a foreground pixel THEN  
Using the current pixel's position  
REPEAT Step 2 UNTIL  
A background pixel is found
4. Store the length of the proposed underline
5. IF the underline length is smaller than half of the word length THEN  
Go back to Step 1  
ELSE  
A proposed underline is found
6. The proposed line's thickness is measured (Section 2.1.1.1).
7. Finally, the underline is removed as per Section 2.1.2.2.

## 3. Implementation and Results

### 3.1 Implementation and Database

The skew detection and underline removal algorithms along with all associated sub-algorithms/techniques were implemented in C and tested on a UNIX platform.

A number of experiments were conducted to test the proposed techniques. All handwritten words from the CEDAR benchmark database [8] were employed for testing purposes. Specifically, all words contained in the "BD/cities" directory of the CD-ROM were used. For testing, 317 words were obtained from the test set.

### 3.2 Experimental Results

The performance of each technique was evaluated by visually inspecting the preprocessed word images. Following baseline estimation, it was found that in 97.8% of cases, ascender and descender strokes were successfully removed. With respect to underline removal, our technique performed satisfactorily in 97.16% of cases. Finally, the skew of each word was correctly detected in 96.21% of cases. Some examples of successful preprocessing attempts may be seen in Figures 2 & 3 below. Some examples of unsuccessfully preprocessed words are shown in Figure 4.

Original Word	Following Underline Removal

Figure 2. Word samples before and after successful underline removal

Original Word	Following Skew Detection and Correction

Figure 3. Word samples before and after successful skew detection and correction

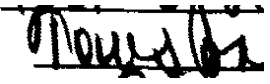
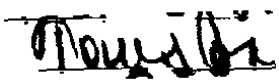

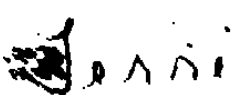



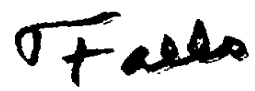


Original Word	Following Preprocessing
	
	
	
	
	

Figure 4. Examples of unsuccessfully preprocessed words

## 4. Discussion

### 4.1 Baseline Detection

As mentioned in Section 3.2, our baseline detection technique performed successfully in a majority of cases. Out of 317 words, the ascenders and descenders were correctly removed save for seven words. In all cases, the reason that the baselines were erroneously detected was due to the fact that the "Peak Line" was incorrectly set. This occurred as a result of a number of characters that contained large horizontal strokes such as the letter "t". In future research, our algorithm may be updated to specifically search for characters with long horizontal strokes in the two upper quarters of each word.

### 4.2 Underline Removal

The process of underline removal posed two major problems. The most obvious problem was a failure to remove the underline. This occurred for only one word, whereby the underline was not satisfactorily removed. Upon examining the word in question, it was found to contain two heavy underlines and an abundance of noise that adversely affected the algorithm.

The second challenge faced by the underline removal technique was to accurately retain strokes in a word that were not meant for removal. This type of error occurred in eight cases. Out of these words, many contained excessive noise that affected the algorithm adversely. The remainder of words was only mildly affected by "over-removal" of foreground pixels.

### 4.3 Skew Detection

The skew detection algorithm was highly successful in measuring the slope for practically all words in the test set aside from twelve. Upon examining each of the failures closely, it was found that for those cases where

the skew was not correctly detected, the words were only mildly tilted and further processing by our system would not be impeded.

## 5. Conclusions and Future Research

New preprocessing techniques used in a system for off-line handwritten word recognition have been presented. Techniques for underline removal and skew detection have been proposed and tested on a benchmark database of handwritten words. Underline removal was successful in 97.16% of cases and word skew was correctly determined in 96.12% of cases.

In future research, the baseline estimation algorithm shall be updated to take into account those words that contain abnormal horizontal strokes. An extra step shall also be included in our word recognition system to remove excessive noise that adversely affects our algorithms.

## References

- [1] M. K. Brown & S. Ganapathy, Preprocessing techniques for cursive script word recognition, *Pattern Recognition*, 16(5), 1983, 447-458.
- [2] A. W. Senior & A. J. Robinson, An off-line cursive handwriting recognition system, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3), 1998, 309-321.
- [3] M. Blumenstein, *Intelligent techniques for handwriting recognition* (PhD Dissertation, Griffith University - Gold Coast, 2000).
- [4] R. M. Bozinovic & S. N. Srihari, Off-line cursive script word recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1), 1989, 68-83.
- [5] G. Kim, V. Govindaraju & S.N. Srihari, Architecture for handwritten text recognition systems, *Advances in Handwriting Recognition* (S.W. Lee, World Scientific Publishing, 1999), 163-182.
- [6] S. Madhvanath, E. Kleinberg & V. Govindaraju, Holistic verification of handwritten phrases, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 1999, 1344-1356.
- [7] G. Dimauro, S. Impedovo, G. Pirlo & A. Salzo, Removing underlines from handwritten text: An experimental investigation, *Progress in Handwriting Recognition*, (A. C. Downton & S. Impedovo, World Scientific Publishing, 1997), 497-501.
- [8] J. J. Hull, A Database for Handwritten Text Recognition, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 16, 1994, 550-554.
- [9] J. R. Parker, *Practical Computer Vision Using C*, (New York: John Wiley & Sons Inc., 1994).