COVER SHEET

Cai, Jinhai and Liu, Zhi-Qiang (2000) OFF-LINE UNCONSTRAINED HANDWRITTEN WORD RECOGNITION. *International Journal of Pattern Recognition and Artificial Intelligence* 14(3):pp. 259-280.

Off-Line Unconstrained Handwritten Word Recognition

Jinhai Cai Zhi-Qiang Liu

Abstract—In this paper, we describe our system for writer independent, off-line unconstrained handwritten word recognition. We have developed a new method to automatically determine the parameters of Gabor filters to extract features from slant and tilt corrected images. An algorithm is also developed to translate 2D images to 1D domain. Finally, we propose a modified dynamic programming method with fuzzy theory to recognize words. Our initial experiments have shown promising results.

keywords—Unconstrained handwritten word recognition, Writer independent, Slant and tilt correction, Gabor filter, Dynamic programming, Fuzzy logic.

1 INTRODUCTION

Automatic recognition of handwriting is important in many applications, for instance, postal address and code reading in postal offices, data acquisition in banks, etc. This topic has gained the considerable attention from industry as well as research community. After intensive research for several decades, many algorithms have been proposed and impressive progress has been made in unconstrained handwritten character recognition. According to recent reports [1,2], the recognition rate of 86.05% to 99.50% has been achieved for handwritten numerals. But, the writer independent, off-line, unconstrained handwritten word recognition still represents a considerable challenge. Depending on different experimental conditions, recognition rate ranging from 42.5% [3] to 92.6% [4] has been reported. Even for writer dependent cursive word recognition [5,6], an average recognition rate of seven writers

is about only 86% for learning files and 76% for testing files [5], which is much lower than that of the human. This is mainly due to three difficulties:

- Off-line recognition deals with two-dimensional images because the dynamic information of strokes is not available;
- Unconstrained handwritten word recognition must deal with all possible writing styles;
- Noise, deformation and different writing tools further deteriorate the performance of the off-line recognizer.

For off-line handwritten word recognition, there are two basic approaches. One is to segment the word into characters or sub-character parts, and then recognize the individual characters with character (or pseudo-character) models [8]. Because many character combinations are not legible, contextual post-processing is performed to detect errors and correct them with the aid of a dictionary [4]. The advantage of this approach is that only few models or references are needed for any words, and the major drawback is that the approach is susceptible to segmental errors. In order to reduce segmental errors, some methods use implicit segmentation techniques. They perform recognition and character segmentation at the same time [11]. However, they cannot totally avoid segmental errors. This is because the overlapping and the interconnection of neighboring characters occur in many handwritten words (see Fig.1). Further, the individual character models ignore the relationship among neighboring characters in a cursive word. For instance, the character "u" in Fig.2(a) & (b) is influenced by their preceding characters.

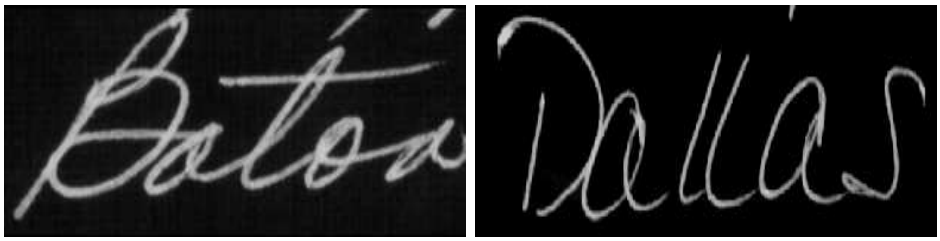


Figure 1: Examples of the overlapping and the interconnection of neighboring characters.

(a) (b)

Figure 2: Examples of characters are influenced by their preceding and following characters.

Another type of methods is the global approach which recognizes a word as a single entity. The global approach can avoid segmental errors, but it needs at least one template or model for each word. As this approach does not deal with characters and uses the relationship among neighboring characters, they are usually considered to be tolerant to the dramatic deformations that affect unconstrained cursive scripts [12]. One major drawback of global methods is that the lexicon can only be updated by training word samples.

Feature extraction is a key step in handwritten word recognition. Many researchers extract features from binarized images by skeletonization. However, when a image is binarized, some important information is lost. Thinning algorithms [13,14] introduce some distortions, such as false skeletal branches. Wang and Pavlidis developed an algorithm [15], which extracts features directly from gray-scale images. Its limitations are that the skeleton is not unit width and the algorithm tends to break connectedness at corners and crosses. Instead, we extract orientation features using Gabor filters that are orientation selective and optimal in joint spatial-spectral information resolution.

2 PRE-PROCESSING

In this section, we describe the pre-processing in our off-line handwritten word recognition system. Pre-processing is designed to reduce noise and variations caused by different writing styles. It consists of slant and tilt correction, baseline finding, image normalization, and

line-width calculation.

2.1 Slant and tilt correction

A. Slant Correction

Word slant is defined as the average slope of word referring to the vertical direction. The word slant is one of main variations in handwriting styles. In order to cope with this variation, we normalize all images to a standard form with no slant [7,8]. The corrected image will produce a more consistent set of features that are used to improve the performance of the system. In addition, slant correction significantly reduces the difficulties encountered in character segmentation that is necessary for segmentation-based techniques.

There are several slant correction algorithms. Bozinovic and Srihari [8] estimate the word slant from binarized images. They remove all horizontal lines by horizontal strip bars and discard all short lines. These strip bars also divide the image into parts. The slope of the word is defined as the average slope of all parts. This algorithm may fail when a word has an obvious ascender or descender. Recently, Buse, Liu, and Caelli [7] proposed to use Fourier transform to convert an image into frequency domain. Because the global orientation of the word is exhibited in its spectrum, the slope of the word can be calculated from the angular histogram of the spectral magnitude image.

In this paper, based on the idea of horizontal line removal from Bozinovic and Srihari [8], we develop a new efficient and robust algorithm to determine the global slope of a word. For a given image, we first binarize the grey-level image using an adaptive threshold method [16], then detect edges. Fig.3 gives an example for edge detection.

However, the edge of image cannot be directly used for slope estimation, as it contains cross points and prominent corners where the orientation of the edge changes dramatically. The cross points are detected according to neighbor numbers of pixels, $N_8(p)$ [13], in the 8-neighborhood. If $N_8(p) > 2$, the nonzero point is a cross point. Corner detection is

(a) (b)

Figure 3: Examples of edge detection. (a) binarized image; (b) edge of the binarized image.

performed using the method [17] proposed by Cheng and Hsu. After deleting cross and corner points, an edge curve is represented by line segments. Now, the global slope of a word can be calculated by

$$slope = \frac{\sum_{\theta_i \in S} l_i \theta_i}{\sum_{\theta_i \in S} l_i}, \tag{1}$$

where, l_i is the segmented line length, θ_i is the angle between the segmented line and the x-axis, and the support, S, is in the range $[30°, 150°]$. The designed support range excludes all horizontal lines. Some results of slant correction are shown in Fig.4 and Fig.5.

B. Tilt Correction

Tilt of a word is defined as the general ascending or descending trend of the writing with respect to x-axis. The tilt correction is necessary to reduce the variation in writing styles. Tilt estimation is based on the method in [7]:

$$tilt = \arg\max_{\alpha \in T} \{\max_{j \in Y} FD_s(\alpha, j)\}, \tag{2}$$

where T is the set of angle values, Y is the range of vertical coordinate, and $FD_s(\alpha, j)$ is the first derivative of smoothed projected density histogram of slant corrected image at the angle of α and the position of $y = j$. The range of support Y is selected from the bottom to the centroid of the image. As tilts in most words do not exceed 10° and the maximum word tilt angle in our database is 13°, we set the allowable tilt range, T, as $[-15°, 15°]$. However, that the horizontal lines will strongly interfere the tilt estimation. This is because they may produce undesired peaks in smoothed projected density histograms. In order to

avoid such interference, we remove all of the horizontal lines by using Gabor filters which will be discussed in the following section. The horizontal line removal criterion is defined as:

$$p(x,y) = \begin{cases} 0, & \text{if } E^{0^\circ}(x,y) \geq 2E^{90^\circ}(x,y), \\ p(x,y), & \text{else,} \end{cases} \tag{3}$$

where $p(x,y)$ is the gray value of image at (x,y), $E^{0^\circ}(x,y)$ and $E^{90^\circ}(x,y)$ are the output energies of Gabor filters with the orientations of 0° and 90° and at the position (x,y). Fig.4 shows the results of tilt detection and correction with or without horizontal line removal.

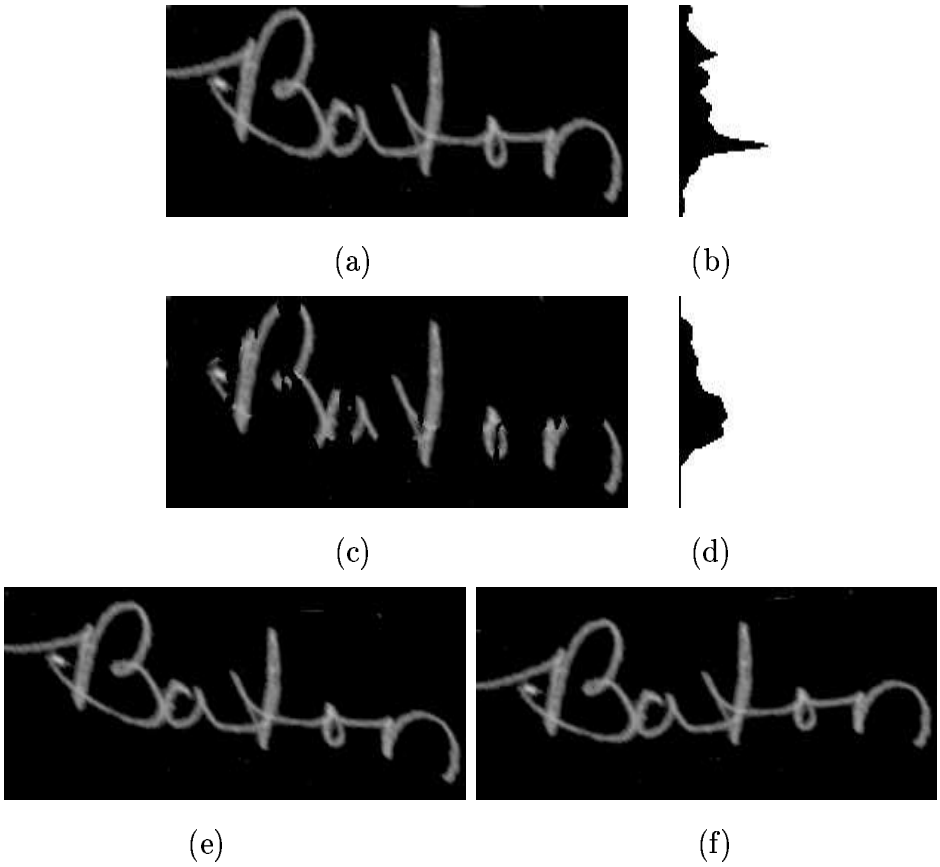


Figure 4: The influence of horizontal line removal on tilt correction. (a) The slant corrected image. (b) The projected density histogram calculated from (a), where the estimated tilt is 3°; (c) The image of (a) after horizontal line removal; (d) The projected density histogram calculated from (c), where the estimated tilt is -5°; (e) The tilt corrected image of (a), where the tilt is estimated from (b); (f) The tilt corrected image of (a), where the tilt is estimated from (d).

As can be seen in this example, there is an 8° difference between two estimated tilts of the same word. Although the procedure described in (2) is simple and effective, the horizontal lines in words may lead this simple procedure to significant errors. Therefore, the horizontal line removal can improve the tilt correction. We estimate word tilts from the images after horizontal line removal. Fig.5 shows more examples of the results of our slant and tilt correction algorithms.

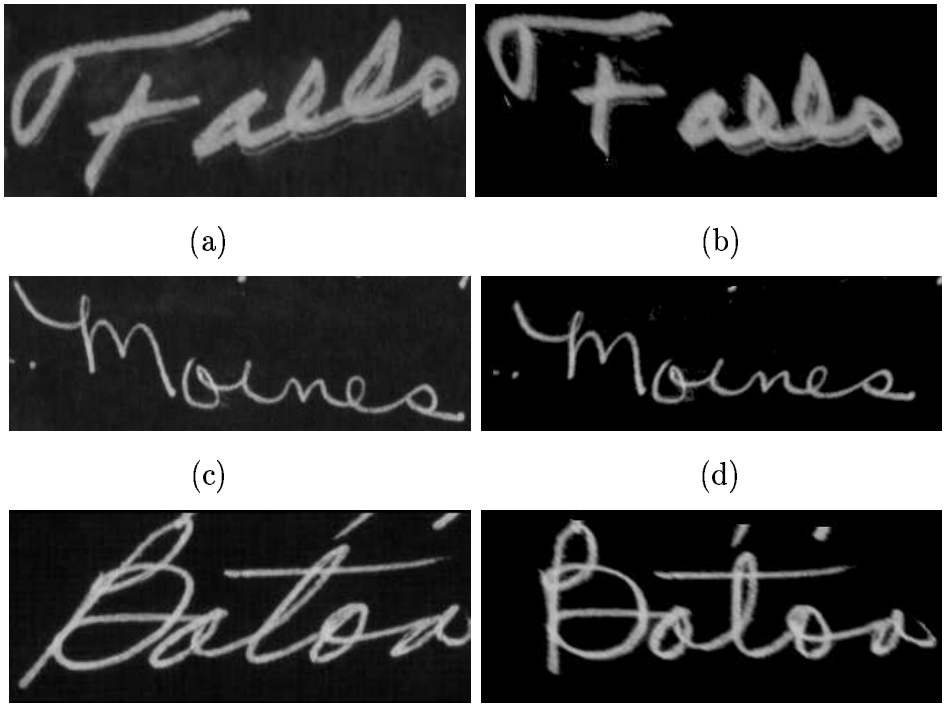


Figure 5: The example of slant and tilt correction. (a), (c) and (e) are the original images and (b), (d) and (f) are the slant and tilt corrected images. In (a), the slant is 61° and tilt is 12.4°; In (c), the slant is 104° and tilt is −3.3°; In (e), the slant is 51° and tilt is 1.7°.

2.2 Baseline finding

We know that spatial positions of word features are very important for word recognition, because this information can greatly narrow the choice of words [8]. In this paper, spatial positions are defined with respect to four baselines: top, upper, lower, and bottom, as illustrated in Fig.6. Bozinovic and Srihari [8] proposed to use thresholds to determine the

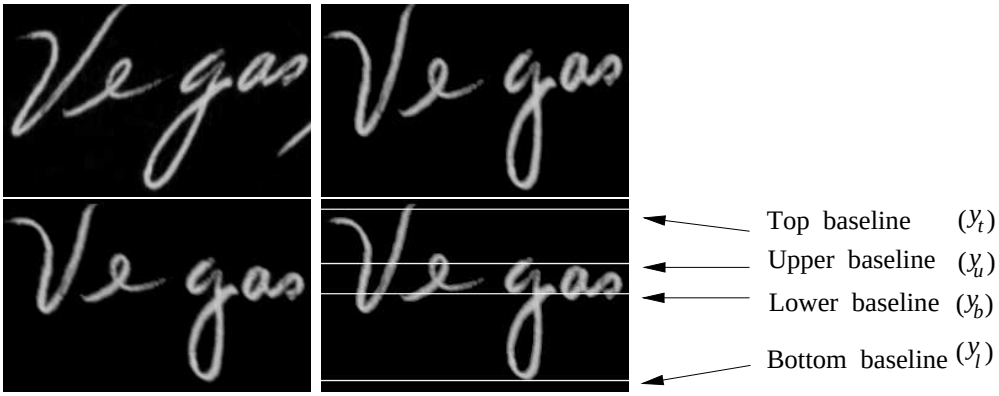


Figure 6: Top left: original image; Top right: slant corrected image Bottom left: slant and tilt corrected image; Bottom right corrected image with four baselines.

shoulders of the density histogram. However, there are two major problems with their algorithm. One is that k of the highest density values have to be discarded. Different words and writing styles produce different number of peaks in density histogram, therefore there is no simple and reliable way to make a good choice for k. Another problem is that there is no obvious shoulder in density histogram for some words after discarding k highest density values. As a result, it is very difficult to adaptively choose suitable thresholds for baseline determination.

In this paper, our proposed algorithm is based on the first derivative of the smoothed vertical density histogram $FD_{st}(0,j)(j \in H)$ of the slant and tilt corrected image after horizontal line removal. $FD_{st}(0,j)(j \in H)$ is equivalent to $FD_s(tilt,j)(j \in H)$ of the slant corrected image after horizontal line removal, and H is the height of the image. As a consequence, we avoid the selection of k by horizontal line removal. The vertical density histogram is slant invariant, but it is very sensitive to ascenders and descenders of words. After tilt correction, we are able to remove this influence. As a result, the shoulder in vertical density histogram becomes more prominent (Fig.4(d)). The position of lower baseline is determined by the gradient of smoothed vertical histogram and the threshold:

$$y_l = \arg\max_{j \in H_l} \{FD_{st}(0,j) | D_{st}(0,j) > T_d\}, \tag{4}$$

where

$$T_d = \min\{\bar{D}_{st}, MD_{st}/2\},$$

$D_{st}(0,j)$ is the smoothed vertical density histogram, T_d is the threshold, H_l is the range from the gravity center of an image down to the first point where $D_{st}(0,j) \leq T_d$, $\bar{D}_{st}$ is the average value of the vertical density function over the range where $D_{st}(0,j) > 0$, and MD_{st} is the highest density value. Another reference line of the main body is the upper baseline which is determined by locating negative peak in the histogram:

$$y_u = \arg\min_{j \in H_u}\{FD_{st}(0,j) | D_{st}(0,j) > T_d\}, \tag{5}$$

where H_u is the range from the gravity center of the image up to the first point where $D_{st}(0,j) \leq T_d$. The top and bottom baselines are found based on the negative and positive peak of $FD(0,j)$ in the outside of the main part.

2.3 Word-box size normalization

Because word sizes differ significantly, this makes it difficult to compare spatial positions of different words. Images are normalized to a standard size of 200×128 pixels. In the normalized image, the lower baseline is located at $y_l = 64$. The scales are determined by the image width and the distances between baselines. The scales, S_x and S_y are calculated by

$$S_x = \frac{200}{IW}$$

$$S_y = \begin{cases} \dfrac{64}{\max\{y_t - y_l, y_l - y_b\}} & \text{if all baselines exist,} \\ \dfrac{64}{\max\{2 \times (y_u - y_l), y_l - y_b\}} & \text{if the top baseline is nonexistent,} \\ \dfrac{64}{y_t - y_l} & \text{if the bottom baseline is nonexistent,} \\ \dfrac{64}{2 \times (y_u - y_l)} & \text{if there are no top and bottom baselines.} \end{cases}$$

where, IW is the image width, y_t, y_u, y_l and y_b are the y-axis coordinates of the top, upper, lower and bottom baselines (Fig.6).

2.4 Line width calculation

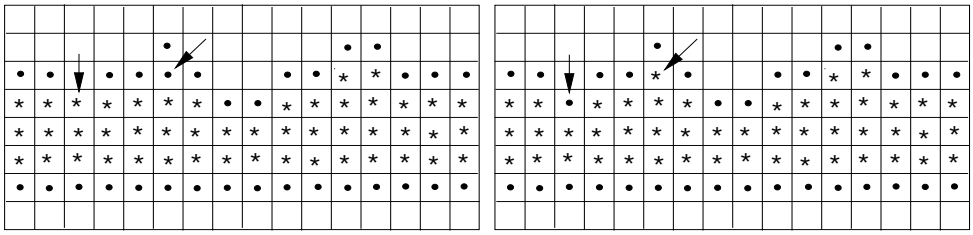
We use the Gabor filter to extract structural features in our experiments. Parameters of the Gabor filter are dependent on the line width of the word. The algorithm for line width estimation is performed on the binarized word image. We assume that the length (L_l) of a line is much larger than the width (W) of the line, $L_l \gg W$. For an ideal line, the sum of its pixels is $S_{um} = L_l \times W$. After deleting its edge points, the sum of its pixels is $S'_{um} = L_l \times (W - 2)$, where $W \geq 2$. The width of the line can be easily calculated by

$$W = \frac{2 \times S_{um}}{S_{um} - S'_{um}}. \tag{6}$$

In real word images, a line can be viewed as an ideal line corrupted by noise. Fig.7 shows an example. In this example, an ideal line is corrupted by adding noise on only one side of its edges. For line width estimation, we proposed two different methods to define edge points. In one method, we define the edge point as a nonzero point and its $N_8(p) \leq 6$, as shown in Fig.7(a). Because not every edge point is real edge point, we call it pseudo-edge point. In the figure, '$\cdot$' represents a pseudo-edge point, and '*' a nonzero pixel with its $N_8(p) \geq 7$. The estimated width of the line is 5. Another method uses the same definition of edge point as that used in slant correction shown in Fig.7(b) where the difference between Fig.7(a) and (b) is indicated by arrows. The estimated width of the line is still 5. For a real image illustrated in Fig.3(a), the average line widths estimated by the two methods are 7.82 and 7.86, respectively. This shows that our method for line width estimation is quite insensitive to noise.

3 GABOR FILTERS AND FEATURE EXTRACTION

The one-dimensional (1D) Gabor filters [26] were developed for signal processing and communication channel analysis. Gabor proved that the Gabor filter family can achieve the theoretical lower bound of joint uncertainty in frequency and time. Daugman [10] extended Gabor's work to 2D case. 2D Gabor filter can reasonably model the 2D receptive field



(a) (b)

Figure 7: Line width calculation. (a) uses the pseudo-edge definition; (b) uses four connected neighbors to define edge point.

profiles of simple cells in mammalian visual cortex. The parameters used in generating Gabor filter can easily control the orientation, spatial extent, frequency and bandwidths of the filter which can be represented as a sinusoidal plane wave of given orientation and frequency within a 2D Gaussian envelop. The Gabor filter, g(x,y), is defined as

$$g(x,y) = \exp\{-\pi(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2})\} \exp\{j2\pi(u_0 x + v_0 y)\}, \tag{7}$$

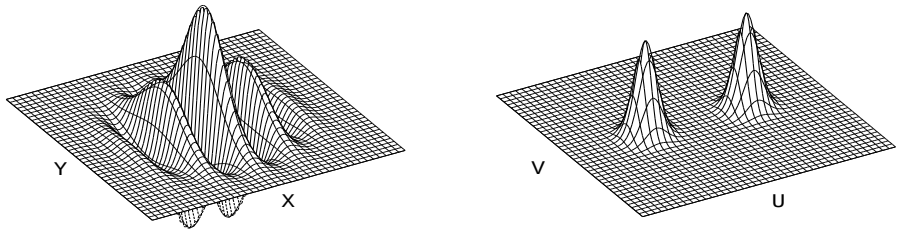
and its 2D Fourier transforms G(u,v) is

$$G(u,v) = \exp\{-\pi[(u'-u_0')^2\sigma_x^2 + (v'-v_0')^2\sigma_y^2]\}, \tag{8}$$

where, $x' = x\cos\phi + y\sin\phi$, $y' = -x\sin\phi + y\cos\phi$, $u' = u\cos\phi + v\sin\phi$, $v' = -u\sin\phi + v\cos\phi$, $u_0' = u_0\cos\phi + v_0\sin\phi$, $v_0' = -u_0\sin\phi + v_0\cos\phi$, $u_0 = f\cos\theta$, $v_0 = f\sin\theta$, $\theta = \phi + 90°$, $j = \sqrt{-1}$ and f is a constant. Fig.8 shows the spatial representation and the 2D frequency response of a Gabor filter with $\phi = 90°$ and $\theta = 0°$. Fig.9 shows the uncertainty relation in spatial and frequency domains.

Recently, the Gabor filter has been used extensively in texture analysis and segmentation, image compression and motion estimation, etc. It can also be used to extract the features of handwritten words. This is because the Gabor Filter has the following major properties:

- It is tunable to specific orientations. This allows us to extract the features of strokes at any possible orientation.



(a) (b)

Figure 8: Gabor filter. (a) Spatial representation of the Gabor filter; (b) Frequency response of the Gabor filter.

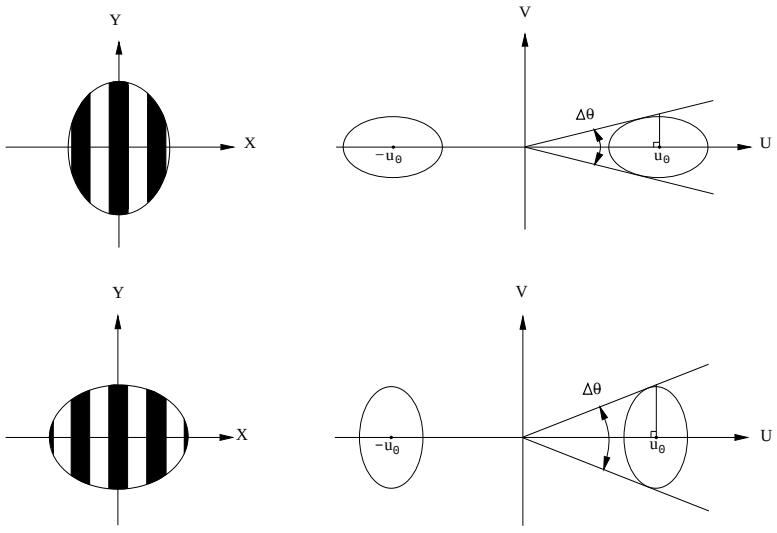


Figure 9: The uncertainty relation in spatial and frequency domains.

- Its orientation bandwidth is adjustable. So, we can use the least number of Gabor filters to achieve the given accuracy of orientation.

- It optimizes the general uncertainty in both spatial and frequency domains. To all 2D linear filters, they are constrained by an uncertainty relation [9]: $(\Delta x)(\Delta y)(\Delta u)(\Delta v) \geq 1/16\pi^2$, where Δx, Δy, Δu and Δv are the position uncertainties and frequency uncertainties, respectively. Therefore, their resolutions simultaneously attainable in spatial and frequency domains are limited. Daugman [9] shows that the Gabor filter can achieve the theoretical limit of the joint 2D resolution regardless of the values of any of its parameters. This means that the Gabor filter is appropriate for tasks requiring simultaneous measurement in both spatial and 2D frequency domains.

- It can extract local information from images. This property is useful to obtain the local orientation of a curve.

- The output of the filter is robust to noise. This is because the Gabor filter uses the information of all pixels within the kernel instead of one pixel.

- It can be implemented by optoelectronic processor at high speed. Because the Gabor filter is not steerable, usually, the operator needs a large number of additions and multiplications. However, the Gabor filtering operations are particularly easy to be implemented by an unusual optical system with VLSI-based processor [18].

3.1 Determination of Gabor filter parameters

Gabor filters have been widely used in image processing and recognition, such as texture segmentation [19], object recognition [20] and image representation [21]. In these applications, the determination of Gabor filter parameters is the central issue. The approach of [20], in which some filter parameters are preset, is not suitable to handwriting feature extraction due to the variation of writing styles. Using a set of Gabor bandpass filters with multiple

orientations and multiple scales [21] is computationally intensive. In order to extract consistent and meaningful structural features from word images, Gabor filters are essential for different sizes and thickness of the writing. Therefore, we proposed a new method to design Gabor filters for handwriting feature extraction. In the following, we will explain how to select the filter parameters in our experiments.

A. Determination of the orientation bandwidth

The number of selected angles for Gabor filters is based on the orientation bandwidth at half maximum response in 2D frequency domain. According to orientation bandwidths of cat cortical simple cells [9], the mean angle covers from 26° to 39°. This means that the least number, n, of Gabor filters covering the range [0°, 180°] is from 5 to 7. For convenience, we choose $n = 8$. In order to reconstruct the original word from extracted parts by Gabor filters, there is $\Delta\theta = \alpha \times 22.5°$ for $n = 8$, where $2 \geq \alpha \geq 1$.

B. Determination of f

The parameter f determines the position of 2D spectral centroids. This parameter will be derived in relation to the average width of lines in the word. Let us consider the worst case, a rectangular pulse line. If the frequency, f, is set too high, the output of the Gabor filter will have two peaks at edges of the rectangular pulse line. In order to produce a single peak in the output for a given line segment, the filter must adapt to different writing tools. Therefore, for a given parameter f, the output of the Gabor filter should satisfy:

$$Out_g(0) \geq Out_g(t) \qquad \frac{W}{2} \geq t > 0, \tag{9}$$

$$Out_g(t_0) \geq Out_g(t_1) \qquad \frac{W}{2} \geq t_1 > t_0 > 0, \tag{10}$$

where

$$Out_g(t) = \int_{-\frac{W}{2}+t}^{\frac{W}{2}+t} \int_{-\infty}^{\infty} \exp\{-\pi(\frac{y^2}{\sigma_y^2} + \frac{x^2}{\sigma_x^2})\} \cos(2\pi f y) dy dx.$$

Because the function $Out_g(t)$ is differentiable, (9) and (10) can be expressed by

$$\frac{dOut_g(t)}{dt} \leq 0 \qquad \frac{W}{2} \geq t > 0.$$

Then, we have

$$\exp\{-\pi\frac{(W/2+t)^2}{\sigma_y^2}\}\cos[2\pi f(W/2+t)] - \exp\{-\pi\frac{(-W/2+t)^2}{\sigma_y^2}\}\cos[2\pi f(-W/2+t)] \leq 0$$

$$\frac{W}{2} \geq t > 0. \tag{11}$$

As (11) must hold regardless of the parameter σ_y, it is easy to obtain

$$\cos[2\pi f(-W/2+t)] \geq \cos[2\pi f(W/2+t)].$$

This gives

$$\sin \pi fW \cdot \sin 2\pi ft \geq 0 \qquad \frac{W}{2} \geq t > 0. \tag{12}$$

A suitable solution for (12) is $f \leq 1/W$ which will produce a single peak in the output of the filter regardless of the values of σ_x and σ_y. This solution can be rewritten as

$$f = \frac{1}{\beta W}, \tag{13}$$

where $\beta \geq 1$. On the other hand, if the selected value of f is too small, the filter may produce one peak in its output of two close lines. Therefore, it is appropriate to take f as $1/W$ for the case of the ideal rectangular pulse line. However, in the database used in our experiments, the lines of words are of gray-scale instead of binary, the equivalent line width is much smaller than the estimated W of binarized words. As a result, β is in the range of [0, 1].

C. Determination of σ_x

The parameter, σ_x, determines the spread of the Gabor filter in ϕ direction. The orientation bandwidth is also mainly determined by σ_x and the frequency f. The relationship between orientation bandwidth (radian) and frequency f and σ_x is illustrated in Fig.9. If $\Delta\theta/2 \ll 180°$, the relationship between them can be approximated by

$$\Delta\theta = \alpha\frac{\pi}{n} \approx 2\arctan(\frac{\Delta F_\phi/2}{f}), \tag{14}$$

where ΔF_ϕ is the 3-dB frequency bandwidth of the filter in v direction when $\phi = 90°$. Applying the conditions of 3-dB frequency bandwidth to (8) results in

$$G(u_0, \frac{\Delta F_\phi}{2})|_{\phi=90°} = exp\{-\pi(\frac{\Delta F_\phi}{2}\sigma_x)^2\} = \frac{\sqrt{2}}{2}. \tag{15}$$

This gives

$$\Delta F_\phi = \frac{\lambda}{\sigma_x}, \tag{16}$$

where $\lambda = \sqrt{2\ln 2/\pi}$. Because the ratio between π and half orientation bandwidth is $\frac{\pi}{\Delta\theta/2} = \frac{2n}{\alpha} \geq n$ and $n = 8 \gg 1$, (14) holds. The parameter, σ_x, can be calculated from the above equations

$$\sigma_x \approx \frac{n\lambda\beta W}{\alpha\pi}. \tag{17}$$

D. Determination of σ_y

The parameter, σ_y, controls the spread of the Gabor filter in the θ direction. Similar to (16) the relationship between σ_y and ΔF_θ is given by

$$\Delta F_\theta = \frac{\lambda}{\sigma_y}, \tag{18}$$

where ΔF_θ is the frequency bandwidth of the filter in u direction when $\theta = 0°$. This relationship is similar to that between the line width and its spectral spread. Therefore, σ_y should be in direct proportion to line width. According to (17), we can infer the parameter constraint

$$\sigma_y = k\sigma_x, \tag{19}$$

where k is a constant. If k is too small, the difference between the outputs of the Gabor filter at different orientations is small. This may result in orientation estimation errors. Furthermore, the outputs of the filter with small k are sensitive to noise. On the other hand, if k is too large, there is a strong interference between outputs of the filter with two close lines. Therefore, the value of k is important to feature extraction. However, we cannot find any clues as to how to determine the coefficient k from (7) and (8). Instead, we select k under the guidance of the positive correlation between space bandwidths or 2D frequency bandwidths

of cat cortical simple cells. It was reported in [9] that the space-domain measurements of k in populations of simple cells usually range between 0.25 and 1. After examining the line extraction results over this range, we find it is appropriate to set $k = 0.75$.

In the above, we have described the method to estimate Gabor filter parameters. The results are consistent with the constraints [9] between space, frequency and orientation bandwidths.

3.2 Feature extraction

The features used in this system are the parameters of word image line segments. For a given line segment having endpoints at pixels $P1(x_1, y_1)$ and $P2(x_2, y_2)$, it is characterized by its orientation θ, length l, and line centroid $c(x_0, y_0)$. These parameters are given as follows.

$$\theta = arc\tan(\frac{y_2 - y_1}{x_2 - x_1}),$$

$$l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2},$$

$$x_0 = \frac{(x_2 + x_1)}{2}, \qquad y_0 = \frac{(y_2 + y_1)}{2}.$$

Extraction of line segments is based on the output of the Gabor filter whose parameters are determined by the method described in the preceding subsection. First, we calculate the power at each point from the complex response of a word image at which the Gabor filter is applied. Then a threshold is used and the oriented lines are obtained from the thresholded image. Fig.18 shows examples of extracted line segments.

4 TRAINING AND RECOGNITION

In our system, two references per word are selected from the training set in the database, and they best represent writing styles of the word. Because word images are 2D, it is difficult to order 2D features in one dimensional domain. Therefore, all the line segments in a word are divided into eight groups according to their orientations. Each group is further divided into three sub-groups based on the four baselines discussed previously (see section 2.2). As a

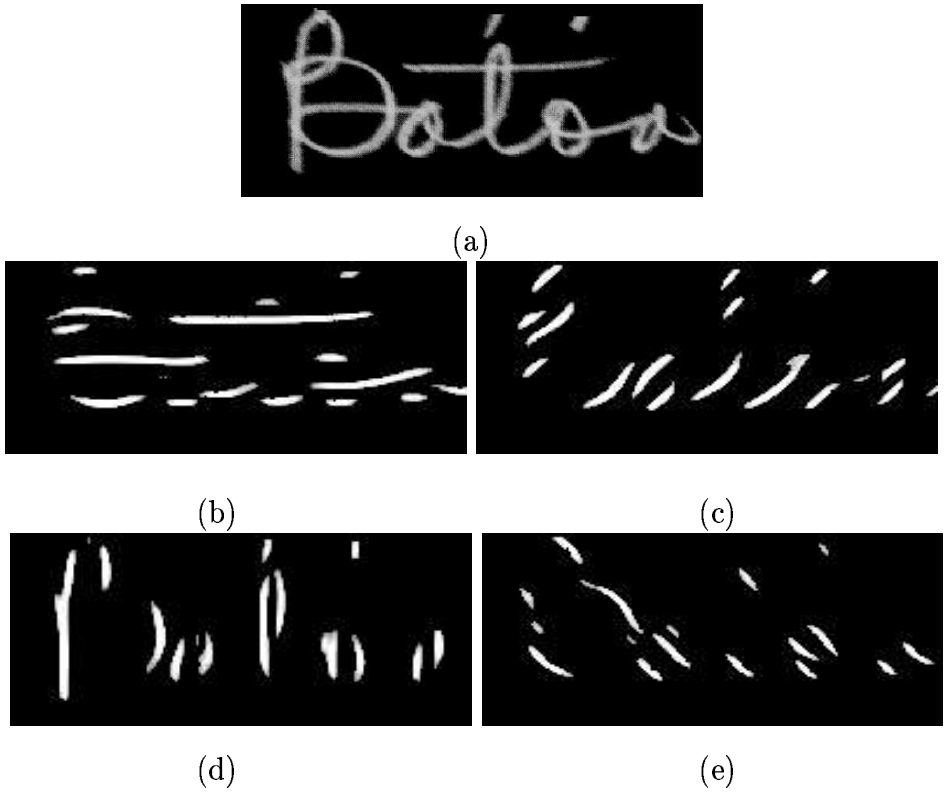


Figure 10: The output energy of Gabor filter for the slant and tilt corrected image in (a). The angle of Gabor filter ϕ is (b) 0°, (c) 45°, (d) 90° and (e) 135°.

result, each sub-group contains only few line segments at a similar angle in a narrow region. Therefore, it is much easier to order them in one dimension by their locations (P_{xy}):

$$P_{xy} = x_0 + \eta(\theta) \times y_0.$$

Here, the lines in one sub-group are arranged with the increscent P_{xy}.

When we try to match one word (I) to another (J), the sub-groups of I are mutually exclusive, while the sub-groups of J are not. This is to avoid the mismatch caused by baseline estimation errors. Global matching is calculated according to each matched line. For a line segment i in a sub-group s of image I, $LI_s(i)$, it may match a line segment j in the sub-group s of image J, $LJ_s(j)$. Their matching relationship can be measured by several methods. For instance, the degree of similarity between two line segments can be defined by coordinate overlapping ratios [22] and high performance has been achieved for handwritten Chinese character recognition. This success is partly due to the small uncertainty in line

segment location within a character. While, for unconstrained handwritten word recognition, it involves the variability of character positions, character shapes, the sizes of word main body and the line segment positions within a character. Therefore, in this paper, the matching relationship is described by a fuzzy logic approach, a weighted additive combiner [10]:

$$\mu_s(i,j) = w_x\mu_x(i,j) + w_y\mu_y(s,i,j) + w_l\mu_l(s,i,j), \tag{20}$$

where w_x, w_y and w_l are weights, μ_x, μ_y and μ_l are the fuzzy membership functions:

$$\begin{aligned} \mu_x &= f_x(x_i - x_j, \sigma_{xj}), \\ \mu_y &= f_y(y_i - y_j, \sigma_{yj}), \\ \mu_l &= f_l(l_i, l_j), \end{aligned} \tag{21}$$

where x_i and x_j are the x-axis coordinators of line i and j respectively, l_i and l_j are their lengths, f_x, f_y and f_l, which are trapezoid-like functions, are defined as follows

$$f_x(x_i - x_j, \sigma_1) = \begin{cases} 0, & |x_i - x_j| > 3\sigma_{xj}, \\ 1, & |x_i - x_j| \le \sigma_{xj}, \\ \frac{3\sigma_{xj} - |x_i - x_j|}{2\sigma_{xj}}, & \text{else}, \end{cases} \tag{22}$$

$$f_y(y_i - y_j, \sigma_{yj}) = \begin{cases} 0, & |y_i - y_j| > 3\sigma_{yj}, \\ 1, & |y_i - y_j| \le \sigma_{yj}, \\ \frac{3\sigma_{yj} - |y_i - y_j|}{2\sigma_{yj}}, & \text{else}, \end{cases} \tag{23}$$

$$f_l(l_i, l_j) = \begin{cases} 1, & C(l_i, l_j) \ge 0.8, \\ 1.25C(l_i, l_j), & \text{else}, \end{cases} \tag{24}$$

where

$$C(l_i, l_j) = \frac{\min\{l_i, l_j\}}{\max\{l_i, l_j\}}.$$

However, the size of word main body in different images may differ greatly. This variation, as illustrated in Fig.11, may result in significant mismatches between two images of the same word. Therefore, we must modify the fuzzy membership functions. Because the variation of x-axis coordinator is small after word size normalization, we modify only the functions of μ_y

and μ_l. In order to achieve size invariance to the word main body, the scale normalization is used in the new definitions of μ_y and μ_l. But some line segments, for instance, the line segments of capitals, are relatively invariant to the size of the main body and are excluded from scale normalization. Specifically, μ_y and μ_l are re-defined as follows.

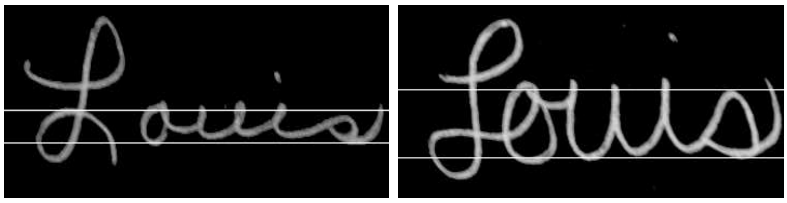


Figure 11: The vast variation of size in word main body.

$$\mu_y(s,i,j) = \begin{cases} f_y(y_i - y_j, \sigma_{yj}) & s = 1, \\ \max\{f_y(y_i - y_j, \sigma_{yj}), f_y[\frac{y_i - y_l}{y_u^I - y_l}(y_u^J - y_l) - (y_j - y_l), \sigma_{yj}]\} & s = 2, \\ \max\{f_y(y_i - y_j, \sigma_{yj}), f_y[\frac{y_l - y_i}{y_l - y_b^I}(y_l - y_b^J) - (y_l - y_j), \sigma_{yj}]\} & s = 3, \end{cases} \tag{25}$$

$$\mu_l(s,i,j) = \begin{cases} f_l(l_i, l_j) & s = 1, \\ \max\{f_l(l_i, l_j), f_l(\frac{l_i}{y_u^I - y_l}, \frac{l_j}{y_u^J - y_l})\} & s = 2, \\ \max\{f_l(l_i, l_j), f_l(\frac{l_i}{y_l - y_b^I}, \frac{l_j}{y_l - y_b^J})\} & s = 3, \end{cases} \tag{26}$$

where we assume that the reference template is image J, y_b^I and y_u^I are the y-axis coordinators of the bottom and upper baselines in image I, y_b^J and y_u^J are these in image J. Dynamic programming is employed to calculate the distance between two words. In the following, we briefly summarize this modified dynamic programming approach:

$$Initialization : D_{sn}(0,j) = 0 \quad 0 \le j \le J_{sn},$$

$$Recursion :$$

$$D_{sn}(i,j) = \min \begin{cases} D_{sn}(i-1,k) + l_i[1 - \mu_s(i,j)] \\ \qquad k \le j, \\ D_{sn}(i-1,j) + l_i\lambda, \end{cases}$$

$$Termination : D_{IJ} = \sum_{n=1}^{8} \sum_{s=1}^{3} D_{sn}(I_{sn}, J_{sn}),$$

where D_{IJ} is the global cost, λ is a cost coefficient of a line segment that does not have a match. In the training phase, all fuzzy membership functions are re-estimated. In order to know which line segment in image J matches the line i in image I, the Viterbi algorithm [23] is used to backtrack the path and to label every line in the training word. The parameters of all labeled lines are transformed into these in reference image J by

$$x'_j(i, I) = x_i, \tag{27}$$

$$y'_j(i, I) = \begin{cases} \frac{y_i - y_l}{y_u^I - y_l}(y_u^J - y_l) + y_l, & s = 2 \text{ and } f_y(y_i - y_j, \sigma_{yj}) < f_y[\frac{y_i - y_l}{y_u^I - y_l}(y_u^J - y_l) - (y_j - y_l), \sigma_{yj}], \\ \frac{y_i - y_l}{y_l - y_b^I}(y_l - y_b^J) + y_l, & s = 3 \text{ and } f_y(y_i - y_j, \sigma_{yj}) < f_y[\frac{y_l - y_i}{y_l - y_b^I}(y_l - y_b^J) - (y_l - y_j), \sigma_{yj}], \\ y_i, & else, \end{cases} \tag{28}$$

and

$$l'_j(i, I) = \begin{cases} \frac{l_i}{y_u^I - y_l}(y_u^J - y_l), & s = 2 \text{ and } f_l(l_i, l_j) < f_l(\frac{l_i}{y_u^I - y_l}, \frac{l_j}{y_u^J - y_l}), \\ \frac{l_i}{y_l - y_b^I}(y_l - y_b^J), & s = 3 \text{ and } f_l(l_i, l_j) < f_l(\frac{l_i}{y_l - y_b^I}, \frac{l_j}{y_l - y_b^J}), \\ l_i, & else. \end{cases} \tag{29}$$

The expectations $\overline{x_j}$ and variances σ'_{xj} of line features can be calculated from $x'_j(i, I)$:

$$\overline{x_j} = \frac{\sum_{I, i \in \mathcal{L}_j} x'_j(i, I)}{N_{\mathcal{L}_j}}, \tag{30}$$

$$\sigma'_{xj} = \sqrt{\frac{\sum_{I, i \in \mathcal{L}_j} [x'_j(i, I) - \overline{x_j}]^2}{N_{\mathcal{L}_j}}}, \tag{31}$$

where $\mathcal{L}_j$ is the set of lines with some matching and $N_{\mathcal{L}_j}$ is the number of line segments in $\mathcal{L}_j$. In the same way, we can obtain $\overline{y_j}$, σ'_{yj} and $\overline{l_j}$. The fuzzy membership functions defined in (22), (25) and (26) are re-estimated by replacing x_j, y_j, l_j, σ_{xj} and σ_{yj} with $\overline{x_j}$, $\overline{y_j}$, $\overline{l_j}$, σ'_{xj} and σ'_{yj}. The matching relationship $\mu_s(i, j)$, which is used in the next training, is obtained according to the re-estimated fuzzy membership functions. Because we train the reference templates with insufficient data, only few training images per word, the estimated features

may be biased. A simple remedy is to use preset constants to ensure that the variances are not unacceptably small:

$$\sigma_{xj} = \max\{\sigma'_{xj}, \delta_x\}, \qquad \sigma_{yj} = \max\{\sigma'_{yj}, \delta_y\}, \tag{32}$$

where δ_x and δ_y are the preset constants.

In the recognition stage, only dynamic programming is used. Backtracking the matching path is not necessary. The global cost is $D_{IJ} + D_{JI}$.

5 EXPERIMENTAL RESULTS

To evaluate our system, a database was extracted from the CEDAR database consisting of USA city name, street name and numerals. This database consists of only USA city names with many writing styles, such as printed, cursive and mixed writing styles. As we know, a pre-classification [24] can be employed to divide all words into several categories with smaller lexicon sizes according to the number of strokes and other simple features. It is relatively simple in practice and theory to differentiate between short and long words. Therefore, our selected database contains 228 words of 5 or 6 letters. Because the writing styles are totally unconstrained and writing tools are unrestricted, we used the pre-processing system to reduce the variations. Line segments are extracted from slant and tilt corrected images by Gabor filters. The similarities between line segments in the template and testing image are measured by fuzzy matching functions (22), (25) and (26) presented in the previous section. The cost of a line segment is defined as the similarity weighted by the line segment length. The global cost is the summation of all sub-group costs obtained by the modified dynamic programming approach.

Our initial experimental results are shown in Table 1, Table 2 and Fig.12, where 115 images with a lexical size of 14 were randomly selected for training and the remaining 113 images were used for testing.

Table 1: Word classification results within the top n positions for testing.

Lexis	positions					The number of samples
	1	2	3	4	5	
Baton	3	-	4	5	-	5
Boise	2	4	-	-	-	4
Dallas	6	8	-	-	9	10
Falls	7	10	-	12	13	16
Haute	3	4	-	-	-	4
Little	4	5	-	6	-	7
Louise	10	13	14	16	-	16
Moines	7	-	9	10	-	10
North	4	5	-	6	-	6
Salem	6	-	-	-	7	7
Sioux	10	-	11	12	-	12
South	4	-	-	-	-	5
Terre	3	-	4	-	-	4
Tulsa	4	5	6	7	-	7

Table 2: Word recognition accuracies for testing

113 testing word images		
first proposal	73	64.6%
among top two proposals	87	77.0%
among top three proposals	94	83.2%
among top four proposals	104	92.0%
among top five proposals	107	94.7%

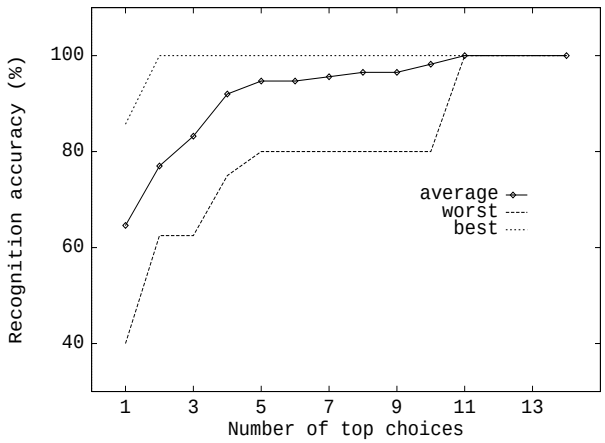


Figure 12: Off-line handwritten word recognition results for testing.

The experimental results were obtained using only 3 to 16 training samples per class (on the average, 7.5 training images per class). Table 1 shows the number of correctly recognized samples within n positions per class over the test set, where $1 \leq n \leq 5$. The correct recognition rates shown in Table 2 are from 64.6% to 94.7% among top 1 to top 5 positions, which are comparable to the recently published results [3,4,8,25], while they used much larger training sets. Fig.12 shows the overall performance with the best and worst cases for handwritten word recognition on the test set. For the best cases, 85.7% and 100% of recognition accuracies for the first and among top two proposals were obtained in our experiments. At the lowest boundary, 40.0% and 100% of correct recognition rates were obtained for the first choice and within eleven choices, respectively.

In the past decade, some impressive results in this field have been reported. But, it is difficult to compare the performance directly between different methods. This is because different systems have been tested on different databases and under different conditions. Nevertheless, the results of the proposed approach, which recognizes off-line unconstrained handwritten words of similar lengths with few training images, are encouraging.

6 CONCLUSIONS

In this paper, we have developed a new pre-processing system to produce more consistent features for handwritten word recognition. In this system, a new simple and efficient method for slope estimation was proposed, which is based on the orientations of non-horizontal line segments. We also proposed a new method to estimate tilt of words using the Gabor filter to remove horizontal lines to avoid their interference. As a result, we were able to obtain better slant and tilt corrected images than that obtained by the pre-processing methods of [7,8].

The Gabor filter is optimal under the general uncertainty principle in joint spatial-spectral information resolution. We have developed a method for the determination of Gabor filter parameters according to the word line width and relationships between orientation bandwidth and frequency bandwidth. Thus, the features extracted by the Gabor filter are insensitive to noise and optimal to the input image in joint spatial-spectral resolution.

We also proposed the modified dynamic programming approach together with fuzzy measures to recognize handwritten words. The performance of our system is encouraging with only few training images.

Improvements of this system are expected with more feature information, such as feature neighborhood and the relationships between sub-groups. It is certain that the recognition rate can be further increased by using a larger training set.

References

[1] C.Y. Suen, C. Nadal, R. Legault, T.A. Mai and L. Lam, Computer recognition of unconstrained handwritten numerals, *Proceedings of the IEEE* 80, 1162-1180 (1992).

[2] I.S.I. Abuhaiba and P. Ahmed, A fuzzy graph theoretic approach to recognize the totally unconstrained handwritten numerals, *Pattern Recognition* 26, 1335-1350 (1993).

[3] T. Paquet and Y. Lecourtier, Recognition of handwritten sentences using a restricted lexicon, *Pattern Recognition* 26, 391-407 (1993).

[4] M.Y. Chen, A. Kundu and S.N. Srihari, Variable duration hidden Markov model and morphological segmentation for handwritten word recognition, *IEEE Trans. on Image Processing* 4, 1675-1688 (1995).

[5] J.C. Simon, Off-line cursive word recognition, *Proceedings of the IEEE* 80, 1150-1161 (1992).

[6] H. Bunke, M. Roth and E.G. Schukat-Talamazzini, Off-line cursive handwriting recognition using hidden Markov models, *Pattern Recognition* 28, 1399-1413 (1995).

[7] R. Buse, Z.Q. Liu and T. Caelli, A structural and relational approach to handwritten word recognition, *IEEE Trans. on Systems Man and Cybernetics*, Part B, Vol.27, 847-861 (1997).

[8] R.M. Bozinovic and S.N. Srihari, Off-line cursive script word recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 11, 68-83, (1989).

[9] J. G. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *Journal of the Optical Society of America* 2, 1160-1169 (1985).

[10] J. M. Mendel, Fuzzy logic systems for engineering: a tutorial, *Proceedings of IEEE* 83, 345-377 (1995).

[11] M. Mohamed and P. Gader, Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 18, 548-554 (1996).

[12] E. Lecolinet and O. Baret, Cursive word recognition: methods and strategies, in *Fundamentals in Handwriting Recognition*, S. Impedovo, Ed., NATO ASI Series F, Vol.124, 235-263, Springer-Verlag (1994).

[13] L. Lam, S.W. Lee and C.Y. Suen, Thinning methodologies–a comprehensive survey, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 14, 869-885 (1992).

[14] M. W. Wright, R. Cipolla and P. Giblin, Skeletonization using an extended Euclidean distance transform, *Image and Vision Computing* 13, 367-375 (1995).

[15] L. Wang and T. Pavlidis, Direct gray-scale extraction of features for character recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 15, 1053-1067 (1993).

[16] N. Otsu, A threshold selection method from grey-level Histograms, *IEEE Trans. on Systems Man and Cybernetics* 9, 63-66 (1979).

[17] F.H. Cheng and W.H. Hsu, Parallel algorithm for corner finding on digital curves, *Pattern Recognition Letters* 8, 47-53 (1988).

[18] H. Urey, W.T. Rhodes, S.P. DeWeerth and T.J. Drabik, Optoelectronic image processor for multiresolution Gabor filtering, *Proceedings of IEEE ICASSP* 6, 3236-3239 (1996).

[19] T.P. Weldon, W.E. Higgins and D.F. Dunn, Efficient Gabor filter design for texture segmentation, *Pattern Recognition* 29, 2005-2015 (1996).

[20] M. Pötzsch, N. Krügert and V. Malsburg, Improving object recognition by transforming Gabor filter responses, *Network: Computation in Neural Systems* 7, 341-347 (1996).

[21] T.S. Lee, Image representation using 2D Gabor wavelets, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 18, 959-971 (1996).

[22] S.L. Chou and W.H. Tsai, Recognizing handwritten Chinese characters by stroke-segment matching using an iteration scheme, in *Character and Handwriting Recognition: Expanding Frontiers*, P.S.P. Wang, Ed., World Scientific Series in Computer Science 30, 175-197 (1991).

[23] A.J. Viterbi and J.K. Omura, *Principles of digital communication and coding*, McGraw-Hill, New York (1979).

[24] D.C. Tseng, H.P. Chiu and J.C. Cheng, Invariant handwritten Chinese character recognition using fuzzy ring data, *Image and Vision Computing* 14, 647-657 (1996).

[25] M.Y. Chen, A. Kundu and J. Zhou, Off-line handwritten word recognition using a hidden Markov model type stochastic network, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 16, 481-496 (1994).

[26] D. Gabor, "Theory of communications," *Journal Of The Institute of Electr. Eng.*, Vol.93, pp.429-457, 1946.

About the Author—JINHAI CAI received his B.S. degree in Physics and M.S. degree in Electronic Engineering from Hangzhou University, China, in 1984 and 1991, respectively. Since 1984, he has been a staff in the Department of Electronic Engineering, Hangzhou University. In 1994, he received the full John Crawford Scholarship and presently he is the PhD Candidate in the Department of Computer Science and Software Engineering, The University of Melbourne. His research interests include speech analysis and processing, handwriting recognition, image processing and intelligent systems.

About the Author—ZHI-QIANG LIU is currently a tenured Associate Professor with Department of Computer Science and Software Engineering, The University of Melbourne. He received a M.A.Sc. degree in Aerospace Engineering from the Institute for Aerospace Studies, The University of Toronto, and a Ph.D. degree in Electrical Engineering from The University of Alberta, Canada. Dr Liu has received a number of prestigious scholarships and fellowships.

Dr Liu is the leader and the Principal Investigator of the Computer Vision and Machine Intelligence Lab (CVMIL) at the Department of Computer Science, the University of Melbourne. Before joining the University of Melbourne, he worked in communications industry as a Principal Engineer.

His research interests include intelligent systems, image processing, computer vision, fuzzy systems, and neural networks. Dr. Liu has over 130 publications and two books in these areas and has received an outstanding paper award by the International Pattern recognition Society. He is an associate editor for IEEE Transactions on Fuzzy Systems, International Journal of Pattern Recognition and Artificial Intelligence, and Computers in Biology and Medicine.