Local Slant Estimation for Handwritten English Words

Yimei Ding, Wataru Ohyama, Fumitaka Kimura
Mie University
1515 Kamihama-cho, Tsu-shi, Mie, Japan
{ding,ohyama,kimura}@hi.info.mie-u.ac.jp

M.Shridhar
The University of Michigan–Dearborn
4901 Evergreen Road, Dearborn, MI, USA

Abstract

This paper describes three methods for local slant estimation, which are simple iterative method, high speed iterative method, and 8–directional chain code method. The experimental results show that the proposed methods can estimate and correct local slant more accurately than the average slant correction.

keywords: *Slant Estimation, Slant Correction, Handwritten English Words*

1 Introduction

Several methods have been proposed and discussed for average slant estimation and correction in the earlier papers [1, 2, 3, 4, 5]. However, analysis of many handwritten documents reveals that slant is a local property and slant varies even within a word. The use of an average slant for the entire word often results in overestimation or underestimation of the local slant. Typically, the slant in a word varies from character to character, as shown in Figure 1. Slant correction is an important factor in applications requiring word recognition, due to the fact that the slant corrected word is more easily segmented into characters (or sub characters). It is possible to achieve more accurate word recognition, if slant correction techniques are based on local slant estimation, rather than the traditional average slant.

2 Local slant estimation and correction

Handwritten words often do not exhibit consistency in stroke orientation. In these cases, local slant will be either over corrected or under corrected by the average slant correction. To solve this problem, local slant estimation using cumulative frequency distribution of chain code is described below.

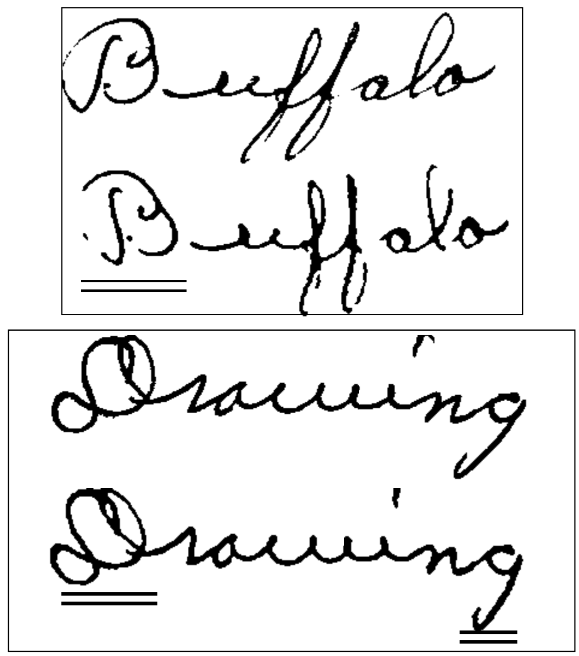


Figure 1. The problem of average slant correction

2.1 Local slant estimation

Local slant estimator is defined as a function of horizontal coordinate x by

$$\theta(x) = \tan^{-1}\left[\frac{n_1(x) - n_3(x)}{n_1(x) + n_2(x) + n_3(x)}\right], \qquad (1)$$

where $n_i(x)(i = 1, 2, 3)$ is the frequency distribution of chain code elements at direction of $i \times 45°$ in $[x - \delta_x, x + \delta_x]$. Figure 2 illustrates the calculation of local slant for a chain code sequence, where $n_1(x) = 4, n_2(x) = 3$, and $n_3(x) = 1$. A parameter δ_x is determined experimentally depending on the input image. The frequency distribution

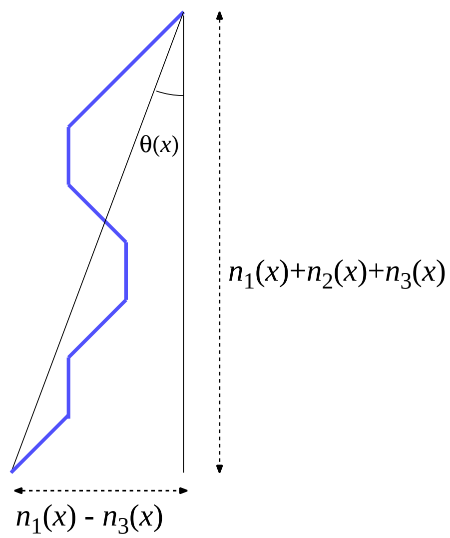


Figure 2. Local slant $\tan\theta(x)$

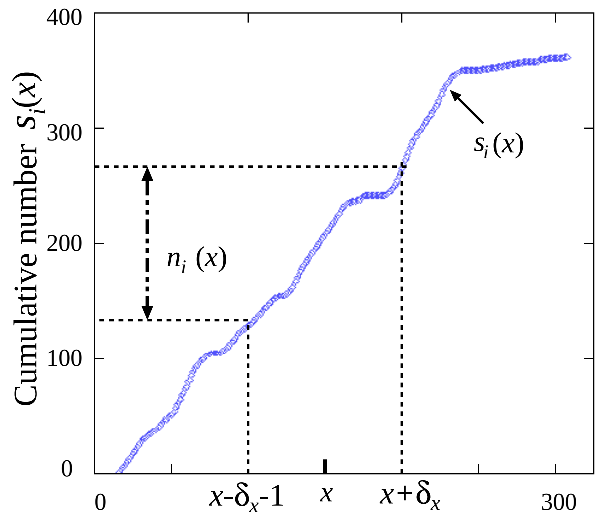


Figure 3. Example of frequency distribution of chain code

$n_i(x)(i = 1,2,3)$ can be calculated as follows:

$$n_i(x) = s_i(x + \delta_x) - s_i(x - \delta_x - 1), \qquad (2)$$

where $s_i(x)$ is cumulative frequency distribution number of chain code elements at direction of $i \times 45°$ in $[0, x]$, as shown in Figue 3.

Then $\tan\theta(x)$ is smoothed by the mean filter of adjacent three pixels. The local slant function $\tan\theta(x)$ for an input image before and after 10 times of smoothing are shown in Figure 4.

2.2 Local slant correction

Shear transformation is then applied to correct the local slant.

$$\begin{cases} x' = x + y\tan\theta(x) \\ y' = y \end{cases} \qquad (3)$$

where (x, y) and (x', y') are the coordinates of before and after the transformation respectively.

Due to the inequality $|\tan\theta(x)| \leq 1$, local slant estimation principlly causes underestimation too, as the average slant estimation does[4]. When the absolute of local slant exceeds $45°$, the estimate is no more correct and valid. To improve the accuracy of local slant estimation, three methods are discussed below.

3 Accuracy improvement of local slant estimation

3.1 Simple iterative chain code method

1. Apply border following to input binary image to detect the chain code.
2. Estimate the local slant using the cumulative frequency distribution of chain code, and smooth the local slant by mean filtering.
3. Apply the shear transformation (3) to the binary image to correct the local slant.
4. Apply 3×3 mean filtering for smoothing jagged contours yielded by the shear transformation.
5. Repeat from 1 until the condition for convergence is satisfied.

In the experiment, estimation accuracy is improved by repeating the process of local slant estimation and correction for 2 or 3 times. However, by the simple iterative chain code method, required computation time for each iteration is not negligible because of the image processing for the local slant estimation, shear transformation and smoothing.

To reduce the processing time, high speed iterative chain code method, by which the local slant correction and the smoothing are applied to the chain code instead of the binary image, is proposed and described below.

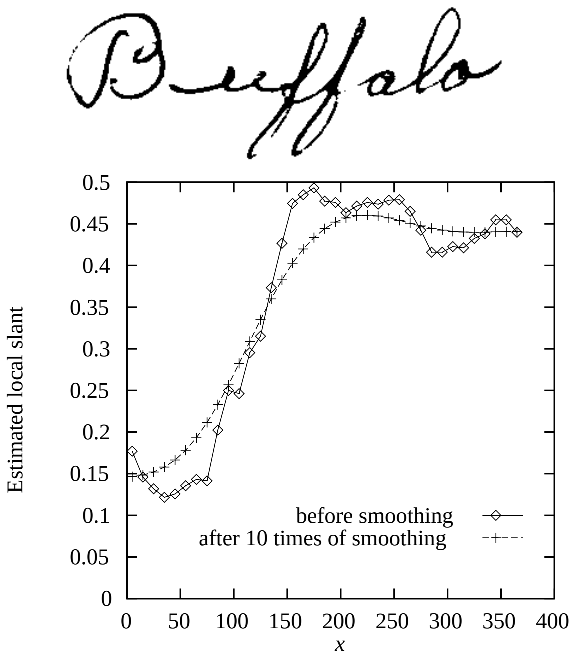


Figure 4. Local slant function $\tan\theta(x)$ before and after smoothing

3.2 High speed iterative chain code method

1) Apply border following to (input) image to detect the chain code.

2) Estimate the local slant using the cumulative frequency distribution of chain code, and smooth the local slant by mean filtering.

3) If the condition for convergence is satisfied, go to step 6).

4) Apply the shear transformation to the chain code to correct the local slant.

5) Apply smoothing operation to the slant corrected chain code, go to 2).

6) Apply the shear transformation to the binary image to correct the total slant.

7) Apply 3×3 mean filtering for smoothing jagged contours yielded by the shear transformation.

The smoothing for the chain code in step 5) is needed because of the same reason described in [4].

3.3 Local slant estimation by 8–directional chain code method

Local slant estimator using 8–directional chain code is given by

$$\begin{aligned} \tan\theta(x) \quad = \quad & [(2n_1(x) + 2n_2(x) + n_3(x)) \\ & - (n_5(x) + 2n_6(x) + 2n_7(x))] \\ & /[(n_1(x) + 2n_2(x) + 2n_3(x)) + 2n_4(x) \\ & + (2n_5(x) + 2n_6(x) + n_7(x))] \end{aligned} \quad (4)$$

where $n_i(x)$ $(i = 1 \sim 7)$ is the frequency distribution of chain elements at direction i in the range $[x - \delta x, x + \delta x]$, and is calculated by formula (2). The term $(2n_1(x) + 2n_2(x) + n_3(x))$ is the sum of horizontal projection of the element 1, 2, 3, $(n_5(x) + 2n_6(x) + 2n_7(x))$ is the sum of horizontal projection of the element 5, 6, 7, and the denominator is the sum of vertical projection of the element 1 to 7.

4 Comparative experiments

Postal word images, e.g. city, street names, from the "bha" database (handwritten address block data collected at Buffalo, New-York) were used in the following experiments.

4.1 Comparison of local slant estimation accuracy

Comparative results of local slant estimation accuracy by simple iterative chain code method, high speed iterative chain code method, and 8–directional chain code method are shown in Figure 5.

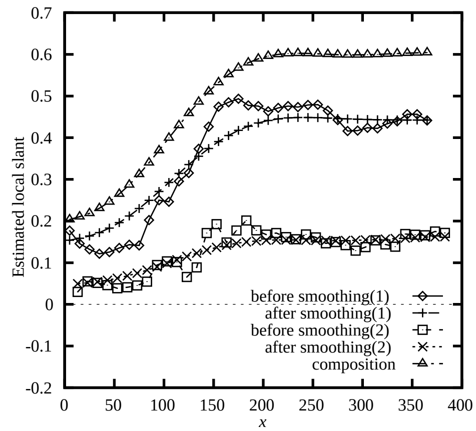
The local slant functions obtained by the simple iterative chain code method are shown in (b). Two of them are the local slant obtained in the first iteration (before and after smoothing), the other two are the local slant obtained in the second iteration (before and after smoothing), and the rest is the composed local slant function of the two iterations.

The local slant functions obtained by the high speed iterative chain code method are shown in (c).

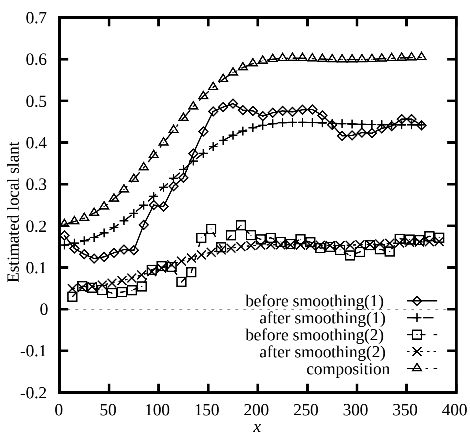
The composed local slant by the simple iterative method, the composed local slant by the high speed iterative method, the local slant by the 8–directional method, and the average slant are shown in (d).

The result (d) shows that the local slant can be estimated more accurately by the three local slant estimation methods when compared with the average slant estimation. The result also shows that the local slant function of non-iterative 8–directional method is close to those of iterative methods with two iterations.

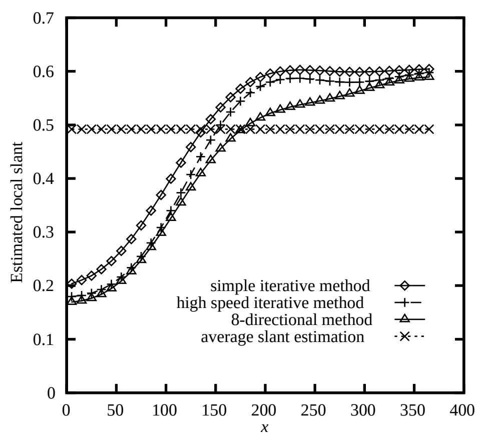
(a) Imput image



(b) Simple iterative method



(c) High speed iterative method



(d) Comparison of three methods and average slant estimation

Figure 5. Estimated local slant $\tan\theta(x)$

4.2 Comparison of local slant correction

Examples of local slant correction by the four methods are shown in Figure 6.

The first row shows an input image, the second row shows the result of average slant correction, the third row to the fifth row show the local slant corrected images by the simple iterative method, the high speed iterative method and the 8–directional method, respectively.

The results show that the local slant correction is more accurate than the average slant correction.

The slant corrected images by the high speed iterative method as well as the 8–directional method are better than those by the simple iterative method, because the shear transformation is applied only once in the high speed iterative method and the 8–directional method, while it is applied twice in the simple iterative method.

4.3 Evaluation and comparison of processing speed

To compare the processing speed of the three local slant estimation methods, the input images are sheared from $-50°$ to $50°$ by every $5°$. Then the processing time for local slant estimation and correction is measured.

Figure 7 shows the relationship between the number of iteration and the average processing time of the three local slant estimation methods for twenty word images. The figure also shows the processing time of simple iterative average slant estimation method [4], for comparison. Used CPU is a hyper SPARC 125MHz.

This figure shows that the processing time of the simple iterative method increases proportional to the number of iteration because the image processing for the local slant estimation, shear transformation and smoothing applied to

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

Figure 6. Examples of local slant correction

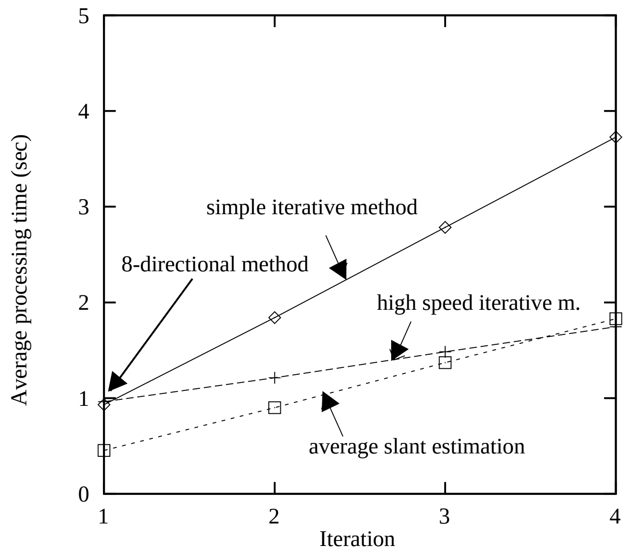


Figure 7. Processing time vs. number of iteration

the binary image is also repeated.

The processing time of the high speed iterative method increases slower than the simple iterative method because the image processing is applied only once.

Finally, the processing time of the 8–directional method is the same as that of the iterative methods of the single iteration.

5 Conclusion

The local slant estimation methods which utilize the local chain code frequency distribution were proposed and shown to be useful to correct slant which varies locally. The estimation accuracy of 8–directional method is close to that of the simple iterative method or the high speed iterative method with two iteration. The processing time of the simple iterative method increases proportional to the number of iteration. While the processing time of the high speed iterative method also increases slowly, the processing time of the non-iterative 8–directional method is the same as that of the iterative methods with a single iteration.

The following studies are remaining as future research topics.

(1) Improvement of local slant estimation for some descending characters such as g, j, y, right upper ligature to next character usually makes slant overestimated.

(2) Further reduction of the processing time of the local slant estimation.

(3) Application of the local slant correction to English word recognition.

(4) Objective evaluation of local slant estimation accuracy.

References

[1] R.M.Bozinovic and S.N.Srihari, ”Off-line Cursive Script Word Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.II, No.1, pp.68–83, January 1989.

[2] F.Kimura, M.Shridhar and Z.Chen, ”Improvements of a Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words”, Proceedings of the Second ICDAR, pp.18–22, October 1993.

[3] D.Guillevic, C.Y.Suen, ”Cursive Script Recognition: A Sentence Level Recognition Scheme”, Proceedings of The Fourth IWFHR, pp.216–223, December 1994.

[4] Y.Ding, F.Kimura, Y.Miyake, M.Shridhar, ”Evaluation and Improvement of Slant Estimation for Handwritten Words”, Proceedings of the Fifth ICDAR, pp.753–756, September 1999.

[5] Y.Ding, F.Kimura, Y.Miyake, M.Shridhar, ”Slant Estimation for Handwitten Words by Directionally Refined Chain Code”, Proceedings of the 7th IWFHR, pp.53–62, September 2000.

Proceedings of the 9th Int’l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 **IEEE**