Non-Uniform Slant Correction for Handwritten Text Line Recognition

Authors
Affiliation
Email

Abstract

In this paper we apply a novel non-uniform slant correction preprocessing technique to improve the recognition of offline handwritten text lines. The local slant correction is expressed as a global optimisation problem of the sequence of local slant angles. This is different to conventional slant removal techniques that rely on the average slant angle. Experiments based on a state-of-the-art handwritten text line recogniser show a significant gain in word level accuracy for the investigated preprocessing methods.

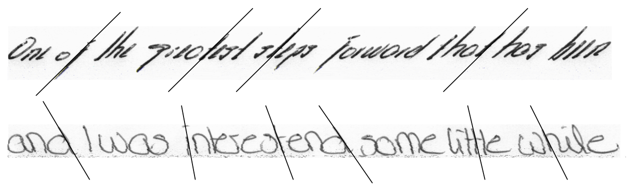

Figure 1. Example of slanted handwriting. The first line shows a uniform slant, whereas the slant angles of the second line are non-uniform.

1. Introduction

After many years of research in handwriting recognition, writer independent recognition of general handwritten text remains a challenging task. Individual writing styles vary greatly and large lexica, required for the recognition of unconstrained text, introduce many similar words that have to be distinguished correctly. As the correct number of words is unknown in advance, an additional source of problems is introduced by segmentation errors. Therefore, it is difficult to achieve a high recognition rate. Depending on the experimental setup, recognition rates between 50% and 80% have been reported for the recognition of handwritten English text [10, 13, 17].

This paper concentrates on the problem of different writing styles which is usually addressed by normalising the input images before they are fed into the recogniser. Specifically, we are interested in writing styles that show a non-uniform slant (see Fig. 1 for an example). These handwritings have been handled rather poorly by previous recognition systems. Particularly in systems where a sliding window is used, non-uniform slant has an adversarial effect on the recognition accuracy.

In conventional slant correction techniques, the average slant angle is estimated and uniformly corrected by a shear transformation. This angle can be estimated by averaging the angles of near-vertical strokes [2, 9], by analysing projection histograms [8, 16], or by statistics of chain-coded stroke contours [3, 4]. These techniques perform well under the assumption that the text line is written with a constant slant. However, the slant angle fluctuates not only between different words, but in some cases at every character in a word (as shown in the second line of Fig. 1). The existence of such handwriting styles is the motivation for a local estimation of slant angles and the corresponding non-uniform correction step. The non-uniform slant correction we apply to handwritten text lines was introduced in [14] for isolated words. The slant correction problem is formulated as a global optimal estimation problem of local slant angels. The optimal sequence that maximises the objective function is searched by a dynamic programming algorithm.

The contribution of this paper is twofold. First, the non-uniform slant correction is applied to entire text lines for the first time. Second, we investigate the impact of the novel slant correction technique on the recognition performance by conducting experiments with a state-of-the-art handwritten text line recognition system. We examine the effect on the recognition accuracy if non-uniform slant correction is applied to the training and testing images, respectively.

The remaining part of the paper is organised as follows. Next, the normalisation steps applied to the images of handwritten text including the non-uniform slant correction are described. Section 3 presents the handwritten text line recognition system. Experiments and results are discussed in Sect. 4 and conclusion are drawn in the last section of the paper.

2. Handwriting Normalisation

To reduce the impact of different writing styles, a handwritten text line image is normalised. In the recognition system used for the experiments described in this paper, the normalisation steps described in [11] are applied. Then, the images are transformed using the non-uniform slant correction introduced in [14].

In the first phase, the images of handwritten text lines are normalised with respect to skew, slant, and baseline position. The skew angle, the slant angle and the position of the upper and the lower baseline are determined globally. The skew correction aligns a line of text with the x-axis of the image and corrects rotations that may have occurred during the writing or the scanning process. The goal of the slant correction is to bring the writing in an upright position. For each line the angle to the vertical axis is corrected. The baseline position correction is an operation based on vertical translation and scaling. The writing is transformed into a standard position and the height is normalised. We refer to [11] for more details about these normalisation steps.

In the second phase the non-uniform slant correction is applied. Assume an M(width) $\times$ N(height) text line image. The problem of the non-uniform slant correction is equivalent to the problem of estimating the local slant angle at each horizontal position $i \in [1, M]$. In order to represent the local slant angle at i, consider a line segment from $(i, 1)$ to (p_i, N), where p_i denote the horizontal position of the upper-end of the line segment. Hereafter, this line segment is called the ith *correction line*. Note that the lower-end of the ith correction line is fixed at $(i, 1)$ while the upper-end is movable horizontally by controlling the variable p_i. Figure 2 shows several correction lines. The slope of the ith correction line (i.e., $\tan^{-1}(p_i - i)/N$) expresses the local slant at position i.

Consequently, the estimation of local slant angles can be treated as a problem of optimising the slope of the correction line at $i = 1, \ldots, M$. Since the slope of the ith correction line is fully controlled by p_i, the non-uniform slant correction problem is finally reduced to an optimisation problem of $p_1, \ldots, p_i, \ldots, p_M$.

Observation of text lines for designing an optimisation criterion reveals that the local slant angles have two properties: (i) long near-vertical strokes tend to exhibit local slant angles clearly, and (ii) the left-to-right transition of the local slant angles is smooth. Thus, roughly speaking, the variable p_i should be optimised so that the slope of the ith correction line is similar to the slopes of long near-vertical strokes around the horizontal position i and the slopes of consecutive correction lines are similar to each other.

According to the above discussion, the non-uniform slant correction problem is defined as follows: Maximise

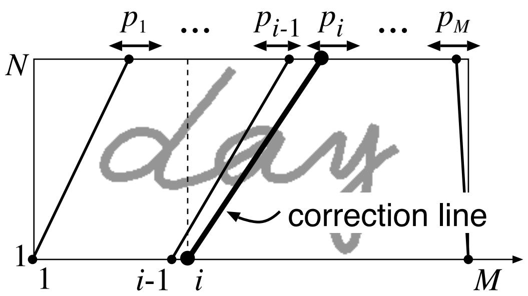

Figure 2. Representation of non-uniform slant angles by correction lines.

$$F(p_1, \ldots, p_i, \ldots, p_M) = \sum_{i=1}^{M} f_i(p_i | p_{i-1}), \quad (1)$$

with respect to $p_1, \ldots, p_i, \ldots, p_M$, subject to

$$0 \leq p_i - p_{i-1} \leq 2, \quad (2)$$

$$|p_i - i| \leq W, \quad (3)$$

where $f_i(p_i|p_{i-1})$ is a function to evaluate a "goodness" of the ith correction line specified by p_i, and W is a positive constant. The details of $f_i(p_i|p_{i-1})$ will be described below. Constraint (2) is imposed to smooth the transition of the local slant angles and constraint (3) to specify the range of local slant angles as $[-\tan^{-1}(W/N), \tan^{-1}(W/N)]$.

The function $f_i(p_i|p_{i-1})$ is defined as a weighted sum of three functions, $s_i(p_i)$, $\gamma(p_i|p_{i-1})$, and $c_i(p_i)$.

- The first function $s_i(p_i)$ becomes larger if a longer stroke is on the correction line. Thus, by the maximisation of $s_i(p_i)$, the slope of the ith correction line will become similar to the slope of a long near-vertical stroke, which will exhibit a local slant angle clearly.

- The second function $\gamma(p_i|p_{i-1})$ evaluates the similarity between the slopes of the ith and $(i-1)$th correction lines. Thus, the maximisation of $\gamma(p_i|p_{i-1})$ smoothes the transition of the local slant angles together with the constraint of (2).

- Several characters, such as "X" and "y", have long strokes with inherent slants. Those inherent slants may be "over-corrected" by the effect of $s_i(p_i)$. The last function $c_i(p_i)$ is introduced to avoid the over-correction. The function $c_i(p_i)$ evaluates the similarity between the local slant angle by p_i and an average slant angle in a neighbourhood of the ith correction

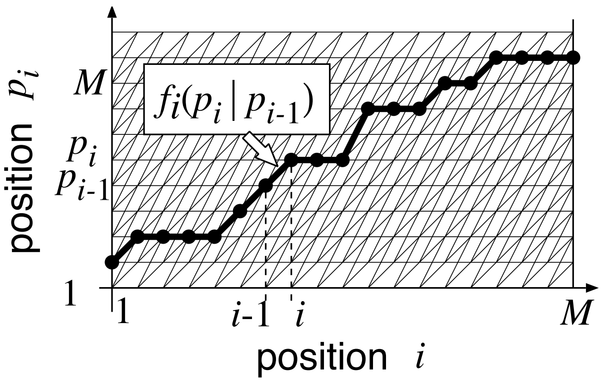

Figure 3. The optimal path problem representing the optimisation problem of $p_1, \ldots, p_i, \ldots, p_M$.

line. The average slant angle is calculated by averaging the slope of the character contour in the neighbourhood. Since the average slant angle is rather insensitive to the above inherent slants, $c_i(p_i)$ is useful to suppress over-corrections. The function also has an effect of stabilising the estimated local slants.

The optimisation problem of $p_1, \ldots, p_i, \ldots, p_M$ can be treated as an optimal path problem as shown in Fig. 3. The path starts from $i = 1$ and ends in $i = N$ on the i—p_i search graph. If the path passes through (i, p_i) next to $(i-1, p_{i-1})$ on the search graph, we have the gain $f_i(p_i|p_{i-1})$. Clearly, the sum of the gains along the path becomes the objective function of (1). The shape and range of the path are regulated by the constraints (2) and (3), respectively. It is well known that the optimal path problem can be solved efficiently by dynamic programming. The algorithm proceeds from $i = 1$ to M (i.e., from left to right in the search graph). At each (i, p_i), the algorithm calculates

$$g(i, p_i) = \max_{0 \leq p_i - p_{i-1} \leq 2} [f_i(p_i|p_{i-1}) + g(i-1, p_{i-1})] \quad (4)$$

where $g(i-1, p_{i-1})$ is the maximum accumulated gain up to $(i-1, p_{i-1})$. This equation implies that the maximum accumulated gain can be calculated recursively. Thus, after the algorithm reaches $i = M$, we can obtain the maximum of the objective function as $\max_{p_M} g(M, p_M)$. The optimal path (i.e., the optimal sequence $p_1, \ldots, p_i, \ldots, p_M$) is obtained by tracing back the selection of p_{i-1} at p_i.

The slant corrected text line image is then obtained by the optimal sequence $p_1, \ldots, p_i, \ldots, p_M$. Specifically, the image is obtained by mapping pixels on the correction line between $(i, 1)$ and (p_i, N) linearly onto the vertical line between $(i, 1)$ and (i, N). This mapping is formally represented as $((p_i - i)j/N + i, j) \mapsto (i, j)$.

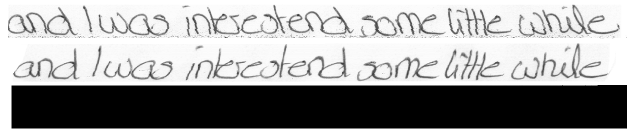

Figure 4. Example of the two different slant correction algorithms. The first line shows the original text line image. The second line shows the image normalised with uniform slant correction whereas the non-uniform slant correction has been additionally applied to the third line.

An example of the slant correction is shown in Fig. 4. The original handwritten text line image, the image after the first phase, and the image after the second phase are shown. Even though the differences between the second and the third line are not striking, we can observe that the handwriting is slightly more upright and uniform after the non-uniform slant correction.

3. Offline Handwritten Text Line Recognition

After these normalisation steps, a handwritten text line is converted into a sequence of feature vectors. For this purpose a sliding window with a width of one pixel is used. The window is moved over the image from left to right, one pixel at each step. Nine geometrical features are extracted at each position of the window. The first three features contain the number of foreground pixels in the window and the first and the second order moment of the foreground pixels. Features four and five contain the position of the upper and the lower contour. The first order derivative of the upper and the lower contour are stored in features number six and seven. The last two features contain the number of vertical black-white transitions and the pixel density between the upper and the lower contour.

In the Hidden Markov Model (HMM) based recognition each character is modelled with a linear HMM. The number of states is chosen individually for each character [17], and twelve Gaussian mixture components are used to model the output distribution in each state. The Baum-Welch algorithm is used for the training of the HMMs, whereas the recognition is performed by the Viterbi algorithm.

A statistical bigram language model supports the Viterbi decoding step. The integration of this language model is optimised on a validation set as described in [17].

4. Experiments and Results

All experiments reported in this section make use of the HMM based recogniser described in Sect. 3. The handwritten text lines used to train, validate, and test the proposed system originate from the IAM database [12].

A writer independent recognition task is considered which implies that none of the writers in the test set is represented in the training or validation set of the system. The training set consists of 6,161 text lines written by 283 writers; 56 writers have contributed 920 text lines to the validation set, whereas the test set contains 2,781 text lines by 161 individuals.

The language model is derived from three different corpora, the LOB corpus [7], the Brown corpus [5], and the Wellington corpus [1]. A bigram language model is built for each of the corpora. These bigram models are then combined linearly with optimised mixture weights [6] to build the final language model.

The underlying lexicon consists of the 20,000 most frequent words that occur in the corpora. The lexicon is not closed over the test set, i.e. there may be out-of-vocabulary words in the test set that do not occur among the 20,000 words included in the lexicon. This scenario is more realistic than a closed lexicon because the texts to be recognised are usually unknown in advance. Our test set contains 6.5% out-of-vocabulary words. This results in a word level accuracy of 93.5% assuming perfect recognition.

The baseline system we use applies all steps of the first prepocessing phase described in Sect. 2, including skew, baseline and uniform slant correction. To test the effect of the proposed non-uniform slant normalisation we conducted two experiments. In the first experiment the trained HMMs of the baseline system are used to recognise the non-uniformly slant corrected lines in the validation and test set. The motivation is that the diversity among the character instances in the training set, which is an important issue in writer independent handwriting recognition [15], is higher if no non-uniform slant correction is applied. In the second experiment the HMMs are trained and tested using the text line images with non-uniform slant correction.

The results on the test set are as follows. If we apply the non-uniform slant correction to the test set only and use the trained HMMs of the baseline system the recognition accuracy increases from 64.5% to 65.8%. Thus, we can benefit from the additional normalisation step to improve the performance of our handwriting recognition system. However, if the non-uniform slant correction is applied to both, training and test images, the word level accuracy drops to 62.7%. Obviously, the normalisation of the training set has a negative impact. A possible explanation is that the diversity of the training samples, which is a important aspect in writer independent recognition, decreases. All differences are statistically significant at the 99% level. The results on the validation and test set are summarised in Tab. 1.

	Validation	Test
Baseline System	69.9%	64.5%
Non-Uniform Slant (Test)	70.2%	65.8%
Non-Uniform Slant (Train & Test)	69.6%	62.7%

Table 1. Word level accuracy results on validation and test set.

5. Conclusions

This paper investigated the use of a non-uniform slant correction technique as an additional preprocessing step for the offline recognition of handwritten lines of text. The motivation for the use of this preprocessing technique lies in the problem that many handwriting styles exhibit a variety of different slant angles within a single line of text or even within individual words. The non-uniform slant correction is formulated as a constrained optimisation problem where the local slant angles represent the variables to be optimised. A dynamic programming based algorithm was then applied to solve this optimisation problem.

Experiments based on a large number of handwritten text lines from the IAM database confirmed the usefulness of the proposed preprocessing method. A statistically significant improvement of 1.3% (absolute) over our already highly optimised baseline recogniser was found when the non-uniform slant correction was applied in addition to our previously existing global normalisation steps. We also found that training the recogniser on non-uniformly normalised text lines did result in a lower word level accuracy. This observation can be explained by our experimental setup (writer independent recognition task) for which an increased variance in the training data seemed to lead to character models that generalise better for new writing styles.

Future work will investigate the behaviour of the new slant correction method in a writer dependent setup. Furthermore, it would be interesting to generalise the concept of local corrections of slant angles to the correction of local skew and base lines as well as the width of characters.

References

[1] L. Bauer. *Manual of Information to Accompany The Wellington Corpus of Written New Zealand English*. Department of Linguistics, Victoria University, Wellington, New Zealand, 1993.

[2] R. M. Bozinovic and S. N. Srihari. Off-line cursive script word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):68–83, 1989.

[3] A. Britto-Jr, R. Sabourin, E. Lethelier, F. Bortolozzi, and C. Suen. Improvement handwritten numeral string recognition by slant normalization and contextual information. In *7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, The Netherlands*, pages 323–332, 2000.

[4] Y. Ding, F. Kimura, Y. Miyake, and M. Shridhar. Accuracy improvement of slant estimation for handwritten words. In *International Conference on Pattern Recognition, Barcelona, Spain*, volume 4, pages 527–530, 2000.

[5] W. N. Francis and H. Kucera. *Brown corpus manual, Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers.* Department of Linguistics, Brown University, Providence, USA, 1979.

[6] J. Goodman. A bit of progress in language modeling. Technical Report MSR-TR-2001-72, Microsoft Research, 2001.

[7] S. Johansson, E. Atwell, R. Garside, and G. Leech. *The Tagged LOB Corpus, User's Manual*. Norwegian Computing Center for the Humanities, Bergen, Norway, 1986.

[8] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis. Slant estimation algorithm for ocr systems. *Pattern Recognition*, 34(12):2515–2522, 2001.

[9] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):366–379, 1997.

[10] G. Kim, V. Govindaraju, and S. Srihari. Architecture for handwritten text recognition systems. In S.-W. Lee, editor, *Advances in Handwriting Recognition*, pages 163–172. World Scientific Publ. Co., 1999.

[11] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[12] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39 – 46, 2002.

[13] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.

[14] E. Taira, S. Uchida, and H. Sakoe. Nonuniform slant correction for handwritten word recognition. *IEICE Transactions on Information & Systems*, E87-D(5):1247–1253, 2004.

[15] T. Varga and H. Bunke. Off-line handwritten word recognition using synthetic training data produced by means of a geometrical distortion model. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(7):1285–1302, 2004.

[16] A. Vinciarelli and J. Luettin. A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 22(9):1043–1050, 2001.

[17] M. Zimmermann, J.-C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):818–821, 2006.