On-Line Recognition of Handwritten Whiteboard Notes: A Novel Approach for Script Line Identification And Normalization

Joachim Schenk and Johannes Lenz and Gerhard Rigoll
Institute for Human-Machine Communication
Technische Universität München
Theresienstraße 90, 80333 München
{schenk, lej, rigoll}@mmk.ei.tum.de

Abstract

In this work we use a previously published approach for script line identification of handwritten whiteboard notes in order to perform skew correction and size normalization of the script trajectory. Arbitrary assignments of sample points to certain script lines are hypothesized and described in a trellis. The normalization is performed by equalizing the script lines and warping the script trajectory accordingly.

In an experimental section we show that the novel normalization achieves a relative improvement of $r = 1.6\,\%$ *in character level accuracy and* $r = 1.4\,\%$ *in word level accuracy compared to a system using standard normalization.*

Keywords: On-line handwriting recognition, whiteboard, normalization, script lines, preprocessing

1. Introduction

In recent years, many publications have addressed the problem of on-line handwriting recognition [9; 14]. While high recognition rates are reported for *isolated word* recognition systems [7], performance considerably drops when it comes to unconstrained handwritten *sentence* recognition. The lack of previous word segmentation introduces new variability. An even more demanding task is the recognition of handwritten whiteboard notes as introduced in [12]. The conditions described in [12] make on-line handwritten whiteboard note recognition difficult.

An important step in any handwriting recognition system is the normalization of the script trajectory. Thereby, writer dependent aspects such as the slant, the skew and the varying sizes in the script are normalized to meet well defined values [9; 12]. A key issue for normalization is the identification of certain script lines (see e. g. [1; 9]) in a line of text as shown in Fig. 1. The top line, the corpus line, the base line,

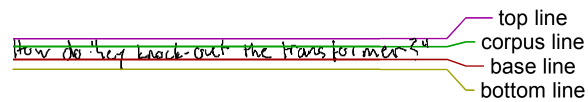

Figure 1. Script lines as e. g. defined in [1; 9]. Script sample taken from IAM-onDB [11].

and the bottom line are (ideally) defined by the top of tall letters (such as "H" and "t"), the top of lower case letters (such as "o" and "w"), the base line points, and the bottom of characters such as "y" and "f" respectively [1]. However, in order to decide if, and in case it does, on which script line a sample point lies, the position and characteristics of each script line must be known. In other words, to find the exact characteristics of the script lines, it must be known which sample points belong to each line [8].

Different approaches for identifying the script lines in a handwritten line of text, aimed at solving the above paradox, have been published. Base lines and corpus lines are described by linear regression lines approximating local minima and local maxima of the trajectory, respectively in e. g. [4]. In [2; 3] the script lines are found by analyzing the profile of the y-projection of the handwritten script. In contrast, all four script lines are approximated as parameterized curves of a second order polynomial in [1; 9]. Thereby, the parameters of the curves are found by fitting a geometrical model to the trajectory by applying the Expectation-Maximization (EM) algorithm [1; 5].

While these approaches seem to work fine for normal handwriting, enhanced algorithms are needed for the variations observed in the script lines of handwritten whiteboard notes. To cope with these variations, in [12], a line of text is heuristically segmented into sub parts and the script lines are separately identified in each of the sub parts.

From: Proc. ICFHR 2008, Montreal, Canada
ISBN 1-895193-03-6

In our previous work [8] we presented a novel method for script line identification. For that purpose sample points which are potential candidates for defining on of the four script lines as depicted in Fig. 1 are found. A trellis is built holding all script line association hypotheses of these points. The path through that trellis leading to least costs is found by applying the Viterbi algorithm and gives the best sample point script line-assignments. Then the script line association is further refined iteratively. Thereby, the resulting script lines may have any characteristics. The sample point script line-assignments are further used to augment a standard feature vector.

In this paper we use the script line-assignment found by the algorithms presented in [8] for skew correction and script size normalization by "equalizing" the script lines. The script lines are forced to run both horizontally and straight. The handwritten script is morphed accordingly.

The remaining paper has the following structure: a brief overview on our baseline system, as well as a short description of the standard preprocessing and the feature extraction used in this paper is given in the next section. In Sec. 3 the novel normalization procedure is described by reviewing the script line identification as introduced in [8] and explaining how the sample point script line-assignment can be used for "equalizing" the script trajectory. The influence of the novel normalization on the word level accuracy is examined in an experimental section (Sec. 4). Conclusions and an outlook are given in Sec. 5.

2. System Overview

In this section we present the preprocessing and normalization used in our baseline system. Then the state-of-the-art features which are extracted from the preprocessed data are briefly summarized. Finally the recognition system based on continuous Hidden Markov Models (HMMs) is roughly described.

2.1. Preprocessing

The x- and y-coordinates as well as the "pressure" p of the handwritten, heuristically line-segmented whiteboard notes are recorded using the eBeam-System as explained in [12]. Afterwards, resampling of the data in order to achieve space equidistant sampling is performed. Then the skew and slant of the script trajectory are corrected using a histogram-based approach as explained in [10] and the corpus and the base lines are estimated similar to [2]. Finally all text lines are normalized to meet a distance of "one" between the corpus and the base line. While this is the only preprocessing for the baseline system, the preprocessed data, especially the extracted base line and corpus line, serve as initialization for the novel normalization approach as presented in Sec. 3.2.

2.2. Feature Extraction

After preprocessing and normalization 24 state-of-the-art *on-line* and *off-line* features [9; 12] are extracted as explained below.

The extracted on-line features are: the pen's "pressure", indicating whether or not the pen touches the whiteboard surface; a velocity equivalent, which is computed *before* resampling is later interpolated according to the resampling factors; the x- and y-coordinate after resampling, whereby the y-coordinate is smoothed by the moving average; the "writing direction", i. e. the angle α of the strokes, coded as $\sin\alpha$ and $\cos\alpha$ and the "curvature", i. e. the difference of consecutive angles $\Delta\alpha = \alpha_t - \alpha_{t-1}$, coded as $\sin\Delta\alpha$ and $\cos\Delta\alpha$.

On-line features which describe the relation between the sample point $\mathbf{s}_t$ to its neighbors ([9; 12]) are (slightly altered if needed): a logarithmic transformation of the "vicinity aspect" v, $\mathrm{sign}(v)\cdot\log(1+|v|)$; the "vicinity slope", i. e. the angle φ between the line $[\mathbf{s}_{t-\tau}, \mathbf{s}_t]$, whereby $\tau < t$ denotes the τ^{th} sample point before $\mathbf{s}_t$, and the bottom line, coded as $\sin\varphi$ and $\cos\varphi$; as well as the "vicinity curliness", the length of the trajectory normalized by $\max(|\Delta x|; |\Delta y|)$. Finally the average square distance to each point in the trajectory and the line $[\mathbf{s}_{t-\tau}, \mathbf{s}_t]$ is given.

The off-line features are: a 3×3 "context map" to incorporate a 30×30 partition of the currently written letter's image, the "ascenders", and "descenders" (i. e. the number of pixels above respectively beneath the current sample point). Further details on the features used can be found e. g. in [16].

2.3. Recognition System

After feature extraction, the handwritten data is recognized by using continuous Hidden Markov Models (HMMs, [15]): each symbol (in this paper: characters) is modeled by one HMM. For comparability, the HMM topology is adopted from [12], using only 32 Gaussian mixtures for approximating the output probabilities. Training of the HMMs is performed by the EM algorithm [5]. Using the Viterbi algorithm the handwritten data are recognized and segmented.

3. Novel Normalization

In this section we describe our novel normalization approach for on-line handwritten whiteboard note recognition. First the script lines are identified by a trellis-based Viterbi search and iteratively refined as explained in [8]. Then the script lines are equalized to run straight and horizontally. By morphing the script trajectory accordingly, the script is both, skew and size normalized. In this paragraph, it is assumed that the handwritten data is preprocessed by the basic steps as explained in Sec. 2.1

3.1. Script Line Identification

As explained in the introduction, the four script lines are defined by certain sample points. However, it is not always clear which sample point lies on a specific script line making the sample point script line-assignment unknown. If the association between sample points and script lines is known, the characteristics of the script lines can be derived. This is the basic principle underlying the approach presented in [8]: deriving the script line characteristics by identifying the sample points lying on the specific line. Thereby, certain sample points become *supporting points* of the script lines. However, each of the N sample points $\mathbf{s}(t)$ contained in the line of text $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_T\}$ may be assigned either to any of the N_l script lines or to no line, leading to $N_\mathrm{tot} = (N_\mathrm{l}+1)^T$ different mappings. One of these mappings contains the "correct" sample point line-assignment. If $T \approx 100$ (this assumption is valid for the database used for experiments in Sec. 4) is assumed, $N_\mathrm{tot} \approx 7.9 \cdot 10^{69}$ different mappings have to be investigated.

In [8] the number of different mappings is lowered by reducing the number of potential sample points lying on a script line. In particular *spatial extreme points* $\mathbf{s}_\mathrm{ext}(n)$, $1 \leq n \leq N_\mathrm{ext}$, with N_ext the actual number of script line defining extreme points, are used for script line definition. After reducing the number of sample points various extreme points script line-assignments are hypothesized. The most likely assignment hypothesis is found by the Viterbi-Algorithm. For further insights, explicit formulation, and refinement of the script line identification see [8]. As a result each script line l is described by the consecutive extreme points $\mathbf{s}_\mathrm{ext}(n) \in \mathcal{L}_l$, where

$$\mathbf{s}_\mathrm{ext}(n) \in \mathcal{L}_l \text{ if } \mathbf{s}_\mathrm{ext}(n) \text{ is assigned to line } l \qquad (1)$$

3.2. Script Line Equalization

After assigning all extreme points to the script lines according to Eq. 1, by properly applying the methods as described [8], the script line can be equalized, i. e. the supporting points $\mathbf{s}_\mathrm{ext}(n) \in \mathcal{L}_l$ of each script line are shifted in order to lie on a horizontal line and meet the same y-position for all text lines. The target heights r_l, $1 \leq l \leq N_\mathrm{l}$ of the script line l is set to $\mathbf{r} = (2, 1, 0, -1)^\mathrm{T}$. By warping each sample point $\mathbf{s}(t) = (x(t), y(t))^\mathrm{T}$ of the script trajectory, which is limited by the script lines according to the shifts of the supporting points, the script trajectory is normalized both in skew and in size.

To perform the mentioned warping the script lines are interpolated between the supporting points. In our paper this is done by a linear interpolation

$$\hat{y}_l(x(t)) = y_l(n) + \frac{y_l(n+1) - y_l(n)}{x_l(n+1) - x_l(n)} \cdot (x(t) - x_l(n)), \qquad (2)$$

where $\hat{y}_l$ is the linear interpolated y-position of script line l at the x-position $x(t)$; $x_l(n)$, $x_l(n+1)$ and $y_l(n)$, $y_l(n+1)$ denote the x- and y-position of the supporting points $\mathbf{s}_\mathrm{ext}(n), \mathbf{s}_\mathrm{ext}(n+1) \in \mathcal{L}_l$ lying closest to $s(t)$. The warped y-position $\tilde{y}(t)$ of each sample point $\mathbf{s}(t)$ of the script trajectory $\mathbf{S}$ is given by

$$\tilde{y}(t) = r_{l_1} + \frac{\hat{y}_{l_2}(x(t)) - \hat{y}_{l_1}(x(t))}{r_{l_2} - r_{l_1}} \cdot (x(t) - \hat{y}_{l_1}(x(t)), \qquad (3)$$

where l_1 and l_2 ($l_1 < l_2$) are the two script lines in between which $\mathbf{s}(t)$ lies. To cope with horizontal distortions due to the vertical warping,

$$s = \frac{1}{T} \sum_{t=1}^{T} \frac{\hat{y}_{l2}(x(t)) - \hat{y}_{l1}(x(t))}{r_{l2} - r_{l1}}, \qquad (4)$$

is derived and horizontal scaling is performed by

$$\tilde{x}(t) = s \cdot x(t), \; 1 \leq t \leq T. \qquad (5)$$

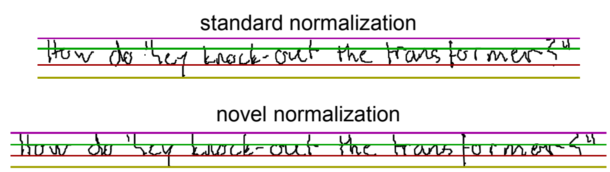

Figure 2. Script trajectory after standard normalization (upper part) and after applying the novel preprocessing (lower part). Script sample taken from IAM-onDB [11].

The result of this warping procedure is shown in the lower part of Fig. 2. After normalization of the script trajectories according to Eqs. 5 and 3 features are extracted as explained in Sec. 2.2.

4. Experimental Results

The experiments presented in this section are conducted on a database containing handwritten heuristically line-segmented whiteboard notes (IAM-OnDB, see [11]). To provide comparability of our results the settings of the writer-independent IAM-onDB-t1 benchmark, consisting of 56 different characters and a 11 k dictionary and provides writer-disjunct sets for training, validation, and test are used.

The first system, our baseline system, uses the preprocessing as explained in Sec. 2.1. The system's parameters are trained on the IAM-onDB-t1's training set until no further improvement evaluated on the combination of both validation sets can be observed. In this stage we achieved a *character level* accuracy on the validation set of $r = 61.2\,\%$ and $A_{\mathrm{b}} = 62.6\,\%$ word level accuracy on the test set of the IAM-onDB-t1 benchmark (see Tab. 1). Then a second system using the preprocessing of the baseline system as initialization for the enhanced normalization (see Sec. 3) is evaluated resulting in a character level accuracy of $a_{\mathrm{new}} = 62.2\,\%$ (validation set) – a relative improvement of $r = 1.6\,\%$, and a word level accuracy of $A_{\mathrm{new}} = 63.1\,\%$ (test set) which is relative improvement of $r = 1.4\,\%$ on the word level. These results are also shown in Tab. 1.

Table 1. Character and word accuracy of three systems (baseline, novel approach, and continuous system [13]).

	A_{b}: baseline	A_{new}: novel normalization	[13]
char. ACC	61.2 %	62.2 %	—
word ACC	62.6 %	63.5 %	65.2 %

However both our systems are outperformed when compared to a recently published continuous system [13] which uses slightly different features and more Gaussians for the continuous HMM based recognition. Some reasons for this drop can be found in [8].

5. Conclusions and Outlook

In this paper we used a recently published method for script line identification (see [8]) for skew correction and script size normalizing for on-line recognition of handwritten whiteboard notes. Therein the script lines of the script trajectory are found via the Viterbi algorithm. Then the script trajectory is normalized by shifting and scaling its sample points in order to meet horizontally running script lines at well defined y-positions. Our experiments show, that a baseline system using standard preprocessing could be outperformed by $r = 1.6\,\%$ relative in character level accuracy and $r = 1.4\,\%$ relative in word level accuracy. However both the baseline and the proposed system were outperformed by a recently published system using a different topology.

In future work, different metrics (such as the ascending slope rather than the absolute y-position of the script lines) will be investigated. We also plan to construct a baseline system with hand annotated script line associations for certain sample points. Additionally the overall training process of the Gaussians will be optimized according to [6].

Acknowledgments

The authors sincerely thank M. Liwicki for providing the lattice for the final benchmark and P. R. Laws for her useful comments.

References

[1] Y. Bengio and Y. Cun, "Word Normalization for On-Line Handwritten Word Recognition", *Proc. of the Int. Conf. on Pattern Recogn.*, pp 409–413, 1994.

[2] R. Bozinovic and S. Srihari, "Off-Line Cursive Script Word Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(1):68–83, 1989.

[3] H. Bunke, M. Roth and E. Schukat-Talamazzini, "Off-Line Cursive Handwriting Recognition using Hidden Markov Models", *Pattern Recogn.*, 28(9):1399–1413, 1995.

[4] T. Caesar, J. Gloger and E. Mandler, "Preprocessing and Feature Extraction for a Handwriting Recognition System", *Proc. of the Int. Conf. on Document Analysis and Recogn.*, pp 408–411, 1993.

[5] A. Dempster, N. Laird and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *J. of the Royal Statistical Society B*, 39(1):1–38, 1977.

[6] S. Günter and H. Bunke, "HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components", *Pattern Recogn.*, 37:2069–2079, 2004.

[7] J. Schenk and G. Rigoll, "Novel Hybrid NN/HMM Modelling Techniques for On-Line Handwriting Recognition", *Proc. of the Int. Workshop on Frontiers in Handwriting Recogn.*, pp 619–623, 2006.

[8] J. Schenk and J. Lenz and G. Rigoll, "Line-Members – a Novel Feature in On-Line Whiteboard Note Recognition", *Proc. of the Int. Conf. on Frontiers in Handwriting Recogn.*, 2008.

[9] S. Jaeger, S. Manke, J. Reichert and A. Waibel, "The NPen++ Recognizer", *Int. J. on Document Analysis and Recogn.*, 3:169–180, 2001.

[10] E. Kavallieratou, N. Fakotakis and G. Kokkinakis, "New Algorithms for Skewing Correction and Slant Removal on Word-Level", *Proc. of the Int. Conf. ECS*, 2:1159–1162, 1999.

[11] M. Liwicki and H. Bunke, "IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard", *Proc. of the Int. Conf. on Document Analysis and Recogn.*, 2:1159–1162, 2005.

[12] M. Liwicki and H. Bunke, "HMM-Based On-Line Recognition of Handwritten Whiteboard Notes", *Proc. of the Int. Workshop on Frontiers in Handwriting Recogn.*, pp 595–599, 2006.

[13] M. Liwicki and H. Bunke, "Combining On-Line and Off-Line Systems for Handwriting Recognition", *Proc. of the Int. Conf. on Document Analysis and Recogn.*, pp 372–376, 2007.

[14] R. Plamondon and S. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey", *IEEE Trans on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[15] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. of the IEEE*, 77(2):257–285, February 1989.

[16] J. Schenk, S. Schwärzler, G. Ruske and G. Rigoll, "Novel VQ Designs for Discrete HMM On-Line Handwritten Whiteboard Note Recognition", *Proc. of the 30th Symposium of DAGM*, pp 229–238, 2008.