

PCA in On-Line Handwriting Recognition of Whiteboard Notes: A Novel VQ Design for Use with Discrete HMMs

Joachim Schenk, Stefan Schwärzler, and Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München
Theresienstraße 90, 80333 München
{schenk, schwaerzler, rigoll}@mmk.ei.tum.de

Abstract

In this work we further evaluate a recently published, novel vector quantization (VQ) design for discrete HMM-based on-line handwriting recognition of whiteboard notes. To decorrelate the features, a principal component analysis (PCA) is applied. The novel VQ design ensures a lossless representation of the pressure information while modeling the statistical dependencies between the pressure and the remaining features. This is necessary because standard k -Means VQ systems cannot quantize this binary feature adequately although they have been decorrelated, which is shown in this paper.

Our experiments show that the new system provides a relative improvement of $r = 2.8\%$ in character level accuracy and a relative improvement of $r = 3.3\%$ in word level accuracy compared to a standard k -means VQ system. Additionally our system is compared and proven to be competitive to a state-of-the-art continuous HMM system yielding a relative improvement of $r = 1.4\%$. A relative improvement of up to $r = 0.8\%$ in word level accuracy can be reported when using decorrelated features compared to a system omitting the decorrelation.

Keywords: Handwriting recognition, whiteboard, vector quantization, discrete HMMs, PCA

1. Introduction

Hidden-Markov-Models (HMMs, [15]) have proven their power for modeling time-dynamic sequences of variable lengths. The property, to allow compensation of the statistical variations in those sequences, is critical for their adoption from automatic speech recognition (ASR), making them quite popular in on-line handwriting recognition [8; 14; 20]. More recently, HMMs have been introduced for on-line handwritten whiteboard note recognition, [10]. One distin-

guishes between continuous and discrete HMMs. In case of continuous HMMs, the observation probability is modeled by mixtures of Gaussians [15]; whereas, in the discrete case the probability computation is a simple table look-up. Vector quantization (VQ) is performed to transform the continuous data to discrete symbols. While in ASR continuous HMMs are becoming widely accepted, it remains unclear whether discrete or continuous HMMs should be used in on-line handwriting [16] and whiteboard note recognition in particular.

In a common handwriting recognition system each symbol (i.e. letter) is represented by one HMM (either discrete or continuous). Words are recognized by combining character-HMMs using a dictionary. While high recognition rates are reported for *isolated word* recognition systems [5], performance considerably drops when it comes to unconstrained handwritten *sentence* recognition [10]. The lack of previous word segmentation introduces new variability and therefore requires more sophisticated character recognizers. An even more demanding task is the recognition of handwritten whiteboard notes as introduced in [10]: in a whiteboard scenario during writing, the writer stands rather than sits and the writing arm does not rest. Therefore additional variation is introduced, such that the baseline cannot be approximated by a simple polynomial of low order. Furthermore it has been observed that size and width of characters and words vary to a higher degree on whiteboards than on tablets. These conditions make the problem of on-line whiteboard note recognition difficult.

In our recent work (see [17]) we started to investigate the use of discrete HMMs for the task of on-line whiteboard note recognition with respect to varying codebook-sizes. While in ASR features have a purely continuous nature, in handwriting recognition continuous features are used as well as discrete or even binary features [10]. As shown in [12] the binary fea-

ture “pressure” is one of the most significant features for recognition. Our experiments in [17] indicate that state-of-the-art vector quantizers are not capable of coding this binary feature properly due to quantization errors. In [17] a novel VQ design which is capable of adequately quantizing this feature is proposed. In addition the role of statistical dependencies between the pressure and the remaining features is pointed out. In this paper we aim at further improving the system presented in [17] by removing the correlation of the features. In order to decorrelate the features a principal components analysis (PCA, as explained e. g. in [18]) is applied.

To that end, the next section gives a brief overview of the recognition system including the necessary pre-processing and feature extraction for whiteboard note recognition. Section 3 reviews VQ as well as discrete HMMs and the incorporation of the PCA-based decorrelation is explained. The VQ-system which adequately handles binary features as introduced in [17] is explained and further enhanced to work with the decorrelated features in Sec. 4. The impacts of varying codebook sizes and the decorrelated features in conjunction with the VQ design are evaluated in the experimental Section (Sec. 5), in which our discrete system is compared to a state-of-the-art continuous system. Finally conclusions and an outlook are given in Sec. 6.

2. System Overview

For recording the handwritten whiteboard data the EBEAM-System¹ is used: a special sleeve allows the use of a normal pen. The sleeve itself sends infrared signals to a receiver mounted on any corner of the whiteboard. As a result the x - and y -coordinates of the sleeve as well as the information, whether or not the tip of the pen touches the whiteboard, the binary “pressure” p , are recorded at a varying sample rate of $T_s = 14 \text{ ms}, \dots, 33 \text{ ms}$. Afterwards the written data is heuristically segmented into lines [10].

The sampled data is preprocessed and normalized in a first step. As the handwritten data is recorded at varying rates, the data is sampled neither in time nor in space equidistantly. As a result, two characters with the same size and style may result in completely different temporal sequences, even if written with the same speed. To avoid this time varying effect, the data is resampled to achieve equidistant sampling in space. Following this, a histogram-based skew- and slant-correction is performed as described in [7]. Finally all text lines are normalized similar to [2].

Afterwards features are extracted from the three-dimensional sample vector $s_t = (x(t), y(t), p(t))^T$ in order to derive a 24-dimensional feature vector $\mathbf{f}_t = (f_1(t), \dots, f_{24}(t))$. The state-of-the-art features for handwriting recognition [6] and recently published new features (partly altered slightly) for whiteboard note recognition [10] used in this paper are briefly listed below and refer to the current sample point \mathbf{s}_t . They can be divided into two classes: *on-line* and *off-line* features. As on-line features we extract f_1 : indicating the pen “pressure”, i. e.

$$f_1 = \begin{cases} 1 & \text{pen tip on whiteboard} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

f_2 : velocity equivalent computed *before* resampling and later interpolated

f_3 : x -coordinate after resampling and subtraction of moving average

f_4 : y -coordinate after resampling and normalization

$f_{5,6}$: angle α of spatially resampled and normalized strokes (coded as $\sin \alpha$ and $\cos \alpha$, “writing direction”)

$f_{7,8}$: difference of consecutive angles $\Delta\alpha = \alpha_t - \alpha_{t-1}$ (coded as $\sin \Delta\alpha$ and $\cos \Delta\alpha$, “curvature”)

On-line features describing the relation between the sample point \mathbf{s}_t to its neighbors as described in [10], and altered in this paper if needed, are:

f_9 : logarithmic transformation of the aspect of the trajectory between the points $\mathbf{s}_{t-\tau}$ and \mathbf{s}_t , whereby $\tau < t$ denotes the τ^{th} sample point before \mathbf{s}_t . As this aspect v (referred to as “vicinity aspect”),

$$v = \begin{pmatrix} \frac{\Delta y - \Delta x}{\Delta y + \Delta x} \\ \Delta x = x(t) - x(t - \tau) \\ \Delta y = y(t) - y(t - \tau) \end{pmatrix},$$

tends to peak for small values of $\Delta y + \Delta x$, we narrow its range by

$$f_9 = \text{sign}(v) \cdot \log(1 + |v|).$$

$f_{10,11}$: angle φ between the line $[\mathbf{s}_{t-\tau}, \mathbf{s}_t]$ and lower line (coded as $\sin \varphi$ and $\cos \varphi$, “vicinity slope”)

f_{12} : the length of trajectory normalized by the $\max(|\Delta x|; |\Delta y|)$ (“vicinity curliness”)

f_{13} : average square distance to each point in the trajectory and the line $[\mathbf{s}_{t-\tau}, \mathbf{s}_t]$

The second class of features, the so-called *off-line* features, are:

f_{14-22} : a 3×3 subsampled bitmap slid along pen’s trajectory (“context map”) to incorporate a 30×30 partition of the currently written letter’s actual image

f_{23-24} : number of pixels above respectively beneath the current sample point \mathbf{s}_t (the “ascenders” and “descenders”)

¹<http://www.e-beam.com>

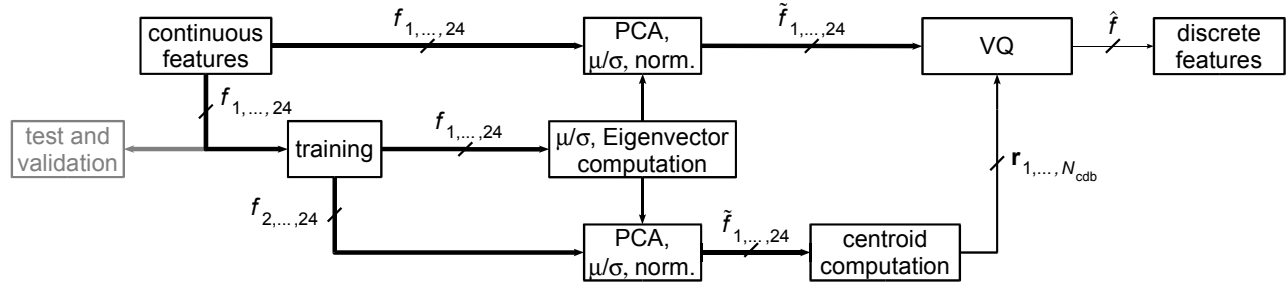


Figure 1. Gaining discrete features \hat{f} from continuous features \mathbf{f} by VQ. Before VQ is performed, the features are decorrelated by applying a PCA and normalized by their means and standard derivations.

3. Vector Quantization and Discrete HMMs

In this section we briefly summarize vector quantization (VQ) and review discrete HMMs.

3.1. Vector Quantization

Quantization is the mapping of a continuous, N -dimensional sequence $\mathbf{O} = (\mathbf{f}_1, \dots, \mathbf{f}_T)$, $\mathbf{f}_t \in \mathbb{R}^N$ to a discrete, one dimensional sequence of codebook indices $\hat{\mathbf{o}} = (\hat{f}_1, \dots, \hat{f}_T)$, $\hat{f}_t \in \mathbb{N}$ provided by a codebook $\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_{N_{\text{cdb}}})$, $\mathbf{c}_k \in \mathbb{R}^N$ containing $|\mathbf{C}| = N_{\text{cdb}}$ centroids \mathbf{c}_i [13]. For $N = 1$ this mapping is called *scalar*, and in all other cases ($N \geq 2$) *vector* quantization (VQ).

Once a codebook \mathbf{C} is generated, the assignment of the continuous sequence to the codebook entries is a minimum distance search

$$\hat{f}_t = \underset{1 \leq k \leq N_{\text{cdb}}}{\operatorname{argmin}} d(\mathbf{f}_t, \mathbf{c}_k), \quad (2)$$

where $d(\mathbf{f}_t, \mathbf{c}_k)$ is commonly the squared Euclidean distance. The codebook \mathbf{C} itself and its entries \mathbf{c}_i are derived from a training set $\mathcal{S}_{\text{train}}$ containing $|\mathcal{S}_{\text{train}}| = N_{\text{train}}$ training samples \mathbf{O}_i by partitioning the N -dimensional feature space defined by $\mathcal{S}_{\text{train}}$ into N_{cdb} cells. This is performed by the well known k -Means algorithm as e.g. described in [3; 4; 13]. As stated in [13], the centroids of a well trained codebook capture the distribution of the underlying feature vectors $p(\mathbf{f})$ in the training data.

The values of the features described in Sec. 2 may be correlated. Also they are neither mean nor variance normalized. First the features are Therefore normalized by their mean ($\mu_i = 0$). Then a PCA using the Eigenvectors of the feature's covariance matrix (as e.g. explained in [18]) is performed on the features in order to achieve decorrelation. Finally the

features are normalized to the standard derivation $\sigma_i = 1$. The overall quantization process is depicted in Fig. 1: first the continuous data is split into a training ($\mathcal{S}_{\text{train}}$), validation (\mathcal{S}_{val}), and test set ($\mathcal{S}_{\text{test}}$), see Sec. 5. The PCA-coefficients, the normalizing factors (μ_j, σ_j), and the centroids \mathbf{c}_i are then calculated from the training set. Finally all data sets are decorrelated, normalized and vector-quantized using the parameters estimated from the training set.

3.2. Discrete HMMs

For handwriting recognition with discrete HMMs each symbol (in this paper each character) is modeled by one HMM. Each discrete HMM i is represented by a set of parameters $\lambda_i = (\mathbf{A}, \mathbf{B}, \pi)$ where \mathbf{A} denotes the transition matrix, \mathbf{B} the matrix of discrete output probabilities corresponding to each possible, discrete observation, and π the initial state distribution [15]. In order to use discrete HMMs the continuous observations $\mathbf{O} = (\mathbf{f}_1, \dots, \mathbf{f}_T)$ are vector quantized yielding discrete observation sequences $\mathbf{o}_i = (\hat{f}_1, \dots, \hat{f}_T)$ as explained in the previous section. Given some discrete training data $\mathbf{o}_i = (\hat{f}_1, \dots, \hat{f}_T)$ the parameters λ_i can be trained with the well known EM-algorithm, in the case of HMMs known as Baum-Welch-algorithm [1]. Recognition is performed by presenting the unknown pattern \mathbf{x} to all HMMs λ_i and selecting the model

$$k_i = \underset{i}{\operatorname{argmax}} p(\mathbf{x}|\lambda_i) \quad (3)$$

with the highest likelihood. In case of word or even sentence recognition this is done by the Viterbi algorithm [19] which also performs a segmentation of the input vector \mathbf{x} .

4. Codebook switching VQ design

Standard k -means VQ cannot adequately model the pen-pressure information as pointed out in [17]

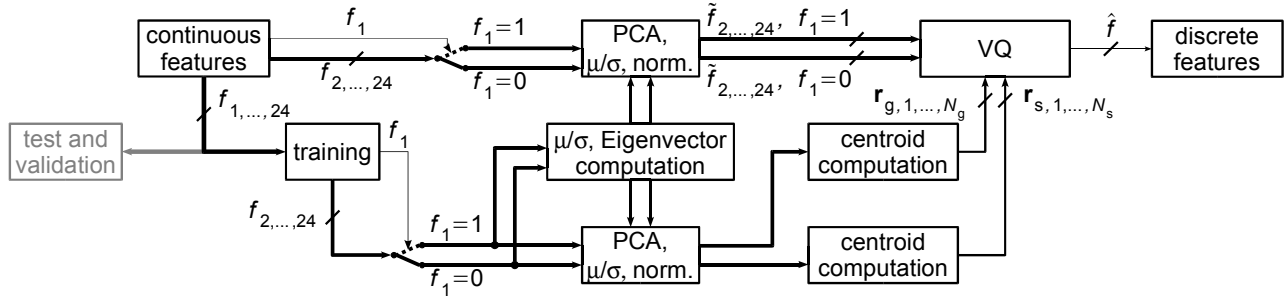


Figure 2. VQ system which models the statistical dependency between f_1 and $f_{2,\dots,24}$ according to Eq. 4 by using two separate codebooks depending on the value of the “pressure” feature according to [17] and enhanced with the PCA-based decorrelation.

and is further shown even for decorrelated features in Sec. 5. However it is stated in [12] that the pressure information is a crucial feature in on-line whiteboard note recognition. To avoid the loss of the pressure information, in [17] we presented a VQ design which two switching codebooks. According to [17] the joint probability $p(\tilde{\mathbf{f}})$ of the features represented by the codebook entries \mathbf{c}_i can be separated into

$$\begin{aligned}
 p(\tilde{\mathbf{f}}) &= p(\tilde{f}_1, \dots, \tilde{f}_{24}) = p(\tilde{f}_2, \dots, \tilde{f}_{24} | \tilde{f}_1) \cdot p(\tilde{f}_1) = \\
 &= \begin{cases} p(\tilde{f}_2, \dots, \tilde{f}_{24} | \tilde{f}_1 < 0) \cdot p(\tilde{f}_1 < 0) & \text{if } f_1 = 0 \\ p(\tilde{f}_2, \dots, \tilde{f}_{24} | \tilde{f}_1 > 0) \cdot p(\tilde{f}_1 > 0) & \text{if } f_1 = 1. \end{cases} \quad (4)
 \end{aligned}$$

applying Bayes’ rule. Depending on the pen’s pressure we switch between two codebooks \mathbf{C}_s and \mathbf{C}_g representing $p(\tilde{f}_2, \dots, \tilde{f}_{24} | \tilde{f}_1 < 0)$ and $p(\tilde{f}_2, \dots, \tilde{f}_{24} | \tilde{f}_1 > 0)$ respectively, during the training and the vector quantization process. By implicitly modeling the pen’s pressure information, this important information is preserved. The number of each codebook’s prototypes (N_s and N_g) may be chosen arbitrarily. The actual number can be derived from the total number of codebook entries $N = N_s + N_g$ and the ratio

$$R = N_g/N_s \Rightarrow N_g = \left\lfloor \frac{N}{1 + 1/R} + 0.5 \right\rfloor, N_s = N - N_g \quad (5)$$

The optimal ratio R for various numbers of N is found by experiment in Sec. 5.

5. Experimental Results

The experiments presented in this section are conducted on a database containing handwritten heuristically line-segmented whiteboard notes (IAM-OnDB²). For further information on the IAM-OnDB, see [9].

²<http://www.iam.unibe.ch/~fki/iamnodb/>

Comparability of the results is provided by using the settings of the writer-independent IAM-onDB-t1 benchmark, consisting of 56 different characters and a 11 k dictionary which also provides well defined writer-disjunct sets (one for training, two for validation, and one for testing). For our experiments the same HMM topology as in [10] is used.

The following three experiments are conducted on the combination of both validation sets, each with seven different codebook sizes ($N = 10, 100, 500, 1000, 2000, 5000, 7500$). For training the vector quantizer as well as the parameters λ_i of the discrete HMMs, the IAM-onDB-t1 training set is used. The results with respect to the actual codebook size N are depicted as *character accuracy* on the left hand side of Fig. 3.

Experiment 1 (Exp. 1): In the first experiment all components of the decorrelated and normalized feature vector ($\tilde{f}_{1,\dots,24}$) are quantized jointly by one codebook. The results shown in Fig. 3 (left) form the baseline for the following experiments. As one can see, the maximum character accuracy $a_b = 62.8\%$ is achieved for a codebook size of $N = 5000$. The drop in recognition performance when raising the codebook size to $N = 7500$ is due to sparse data [15].

Experiment 2 (Exp. 2): To prove that the binary feature f_1 is not adequately quantized by standard VQ, independent of the number of centroids and even after decorrelation, all features except the pressure information ($f_{2,\dots,24}$) are quantized jointly for the second experiment. As Fig. 3 left shows, slight degradation in recognition performance compared to the baseline can be observed. In fact both codebook size-ACC curves run almost parallel. The peak rate of $a_r = 62.7\%$ is again reached at a codebook size of $N = 5000$, which equals a relative change of $r = -0.2\%$. This rather surprising result (in [12] pres-

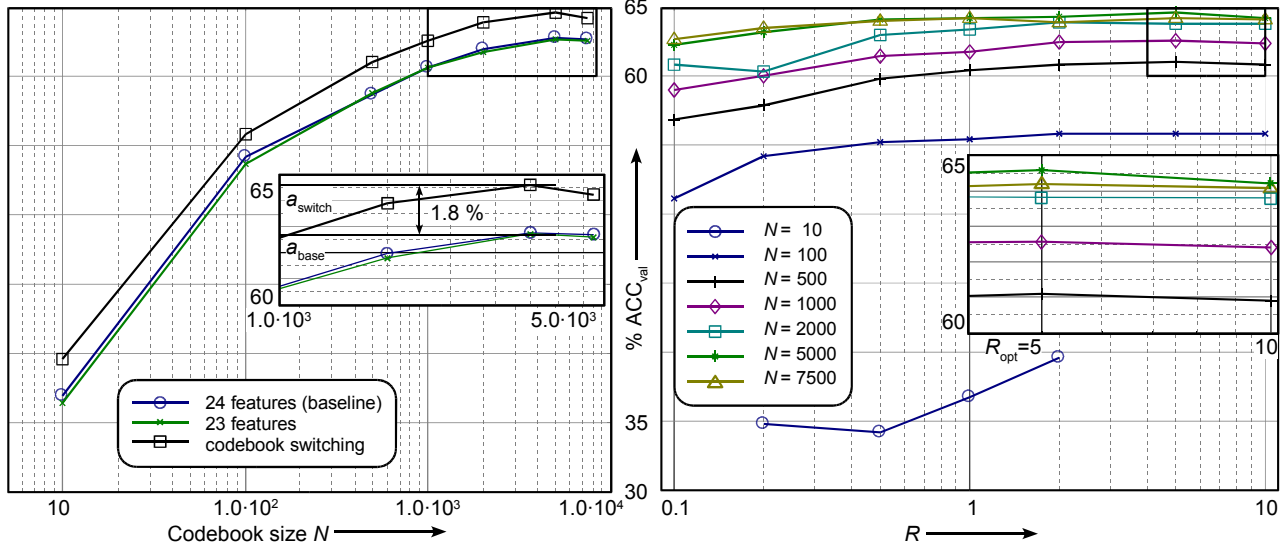


Figure 3. Evaluation of different systems’ character accuracy with respect to the codebook size N (left), character accuracy for different codebook sizes and varying ratios $R = N_g/N_s$ for the VQ design using codebook-switching (right).

sure is assumed to be a relevant feature in on-line whiteboard note recognition) confirms our findings presented in [17].

Experiment 3 (*Exp. 3*): In the last experiment the performance of the novel VQ system as introduced in [17] and further enhanced in Sec. 4 is evaluated. The optimal value of $R = N_g/N_s$ is found by experiment. Investigating the right hand side of Fig. 3 reveals the optimal values for R for arbitrary codebook sizes. Finally the results are shown on the left hand side of Fig. 3 with respect to the codebook size N for the optimal values of R . The highest character accuracy of $a_{\text{sw}} = 64.6\%$ is found for $N = 5000$ and $R_{\text{opt}} = 5$, which yields (according to Eq. 5) $N_s = 833$ and $N_g = 4167$ for the codebooks \mathbf{C}_s and \mathbf{C}_g . Compared to the baseline system, this is a relative improvement of $r = 2.8\%$ ($\Delta R = 1.8\%$ absolute).

In order to prove the competitiveness of the system presented in this paper, the parameters and models which delivered the best performing systems in the previous experiments are taken to perform word level recognition on the test set of the IAM-onDB-t1 benchmark and are compared to a state-of-the-art continuous recognition system as presented in [11] as well as to our results presented in [17] where the decorrelation is omitted. The baseline system, using a standard VQ and coding all features jointly, achieves a word accuracy of $A_b = 63.9\%$. As expected from the character accuracy of the previous experiments (*Exp. 2*), the omission of the “pressure” information has lit-

tle influence on the word level accuracy: $A_r = 63.8\%$ can be reported in this case, describing a drop of $r = -0.2\%$ relatively compared to the baseline system. An absolute word accuracy of $A_{\text{sw}} = 66.1\%$ can be achieved by using the codebook-switching design (*Exp. 3*) which is a relative improvement of $r = 3.3\%$ compared to the baseline. This system even outperforms the continuous system presented in [11] by $r = 1.4\%$ relative.

Additionally Tab. 1 shows some of our results presented in [17] where we used the same features and system settings as in this paper, however with out the PCA. In all experiments the decorrelation of the features leads to an improvement. A relative improvement of $r = 0.8\%$ can be reported in case of the novel VQ design when the features are decorrelated. This may have several reasons. A reasonable explanation is that the underlying VQ estimates better codebooks when the data is decorrelated.

6. Conclusion and Outlook

In this paper we extended a recently published VQ-design scheme (see [17]) for on-line whiteboard note recognition. We normalized and decorrelated the features using a PCA approach. Our experiments with a common VQ system show that the binary pressure information is not adequately quantized regardless of the codebook size even when the features are decorrelated. To overcome this problem, the PCA approach

Table 1. Final results for the experiments 1, . . . , 3 presented in this paper, the corresponding results without decorrelated features as presented in [17], and one different, continuous system [11].

system	with PCA <i>Exp 1</i> (A_b)	wo. PCA <i>Exp 1</i> in [17]	with PCA <i>Exp 2</i> (A_r)	wo. PCA <i>Exp 2</i> in [17]	with PCA <i>Exp 3</i> (A_{sw})	wo. PCA <i>Exp 4</i> in [17]	continuous system [11]
word level accuracy	63.9%	63.5%	63.8%	63.2%	66.1%	65.6%	65.2%

was combined with a novel VQ design recently published in [17] which models the pressure information without any loss. The statistical dependency between the “pressure” and the remaining features is taken into account by using two arbitrary codebooks. The main parameters for this second system are the ratio $R = N_e/N_s$ of the two codebook sizes as well as the total number N of codebooks. Both parameters have been optimized on a validation set by means of maximum character accuracy. The best performing combination led to a relative improvement of $r = 3.3\%$ in word level accuracy on an arbitrary test set, compared to a common VQ system. In comparison to a recently published continuous system, a slight relative improvement of $r = 1.4\%$ can be reported, illustrating the competitiveness of our system. In addition it has been shown that the decorrelation of the features leads to an improvement of up to $r = 0.8\%$ relative.

In future work we plan to extend the approaches presented in this paper to other binary and discrete features commonly used in on-line whiteboard note recognition as well as to investigate the use of multiple stream HMMs as e. g. explained in [16].

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] L. Baum and T. Petrie, “Statistical Inference for Probabilistic Functions of Finite State Markov Chains”, *Annals of Mathematical Statistics*, 37:1554–1563, 1966.
- [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] E. Forgy, “Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications”, *Biometrics*, 21:768–769, 1965.
- [4] R. Gray, “Vector Quantization”, *IEEE ASSP Magazine*, pp 4–29, April 1984.
- [5] 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.
- [6] S. Jaeger, S. Manke, J. Reichert and A. Waibel, “The NPen++ Recognizer”, *Int. J. on Document Analysis and Recogn.*, 3:169–180, 2001.
- [7] 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.
- [8] R. Koehle and T. Matsumoto, “Pruning Algorithms for HMM On-Line Handwritten Recognition”, *Technical Report of IEICE, PRMU97-5*, pp 33–39, 1997.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] M. Liwicki and H. Bunke, “Feature Selection for On-Line Handwriting Recognition of Whiteboard Notes”, *Proc. of the Conf. of the Graphonomics Society*, pp 101–105, 2007.
- [13] J. Makhoul, S. Roucos and H. Gish, “Vector Quantization in Speech Coding”, *Proc. of the IEEE*, 73(11):1551–1588, November 1985.
- [14] K. S. Nathan, J. R. Bellegarda, D. Nahamoo and E. J. Bellegarda, “On-Line Handwriting Recognition Using Continuous Parameter Hidden Markov Models”, *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing*, 5:121–124, 1993.
- [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] G. Rigoll, A. Kosmala, J. Rottland and C. Neukirchen, “A Comparison between Continuous and Discrete Density Hidden Markov Models for Cursive Handwriting Recognition”, *Proc. of the Int. Conf. on Pattern Recogn.*, 2:205–209, 1996.
- [17] J. Schenk, S. Schwärzler, G. Ruske and G. Rigoll, “Novel VQ Designs for Discrete HMM On-Line Handwritten Whiteboard Note Recognition”, *DAGM, accepted for presentation*, 2008.
- [18] M. Turk and A. Pentland, “Eigenfaces for Recognition”, *J. of Cognitive Neuroscience*, 3(1), 1991.
- [19] A. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm”, *IEEE Transactions on Information Theory*, 13:260–267, 1967.
- [20] H. Yasuda, T-Talahasjo and T. Matsumoto, “A Discrete HMM For Online Handwriting Recognition”, *Int. J. of Pattern Recogn. and Artificial Intelligence*, 14(5):675–688, 2000.