Novel VQ Designs for Discrete HMM On-Line Handwritten Whiteboard Note Recognition

Joachim Schenk, Stefan Schwärzler, Günther Ruske, and Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München
80290 Munich, Germany
{schenk,schwaerzler,ruske,rigoll}@mmk.ei.tum.de

Abstract. In this work we propose two novel vector quantization (VQ) designs for discrete HMM-based on-line handwriting recognition of whiteboard notes. Both VQ designs represent the binary pressure information without any loss. The new designs are necessary because standard k-means VQ systems cannot quantize this binary feature adequately, as is shown in this paper.

Our experiments show that the new systems provide a relative improvement of $r = 1.8\,\%$ in recognition accuracy on a character- and $r = 3.3\,\%$ on a word-level benchmark 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-based system yielding a slight relative improvement of $r = 0.6\,\%$.

1 Introduction

Hidden-Markov-Models (HMMs, [1]) have proven their power for modeling time-dynamic sequences of variable lengths. HMMs also compensate for statistical variations in those sequences. Due to this property they have been adopted from automatic speech recognition (ASR) to the problem of on-line (cursive) handwriting recognition [2], and have more recently be applied to for on-line handwritten whiteboard note recognition [3]. One distinguishes between continuous and discrete HMMs. In case of continuous HMMs, the observation probability is modeled by mixtures of Gaussians [1], whereas for discrete HMMs the probability computation is a simple table look-up. In the latter case vector quantization (VQ) is performed to transform the continuous data to discrete symbols. While in ASR continuous HMMs are increasingly accepted, it remains unclear whether discrete or continuous HMMs should be used in on-line handwriting [4] and whiteboard note recognition in particular.

In a common handwriting recognition system each symbol (i. e. letter) is represented by one single-stream 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 recognition of whole unconstrained handwritten *text lines* [3]: 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 [3]. The conditions described in [3] contributes to the characterization of the problem of on-line whiteboard note recognition as “difficult”.

In this paper, we discuss the use of discrete single-stream 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 general, in handwriting recognition continuous features are used as well as discrete or even binary features [3]. As shown in [6] the binary feature “pressure” is one of the most significant features for recognition. Our experiments indicate that state-of-the-art vector quantizers are not capable of coding this binary feature properly due to quantization error. In this paper, we therefore introduce two novel VQ designs which are capable of adequately quantizing this feature.

To that end, the next section gives a brief overview of the recognition system including the necessary preprocessing and feature extraction for whiteboard note recognition. Section 3 reviews VQ as well as discrete HMMs. Two novel VQ systems are introduced in Sec. 4 in order to handle binary features. The impacts of varying codebook sizes and the novel VQ designs 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 discussion are presented in Sec. 6.

2 System Overview

For recording the handwritten whiteboard data the eBeam-System[1] is used. A special sleeve allows the use of a normal pen. This sleeve 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 hits the whiteboard, the binary “pressure” p, are recorded at a varying sample rate of $T_s = 14\,\text{ms}, \ldots, 33\,\text{ms}$. Afterwards, the written data is heuristically segmented into lines [3].

The sampled data is preprocessed and normalized as a first step. In the subsequent feature extraction step, 24 features are extracted from the three dimensional data vector (sample points) $\mathbf{s}_t = (x(t), y(t), p(t))^{\mathrm{T}}$. Then, VQ with varying codebooks and codebook sizes is performed. Finally the data is recognized by a classifier based on discrete HMMs.

2.1 Preprocessing and Feature Extraction

As mentioned above, the data consists of individual text lines recorded at varying sample rates. Therefore 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

[1] `http://www.e-beam.com`

sampling. Following this, a histogram based skew- and slant-correction is performed as described in [7]. Finally all text lines are normalized such that there is a distance of "one" between the upper and lower scriptlines.

Afterwards features are extracted from the three dimensional sample vector $s_t = (x(t), y(t), p(t))^{\mathrm{T}}$ in order to derive a 24-dimensional feature vector $\mathbf{f}_t = (f_1(t), \ldots, f_{24}(t))$. The state-of-the-art features for handwriting recognition and recently published new features (altered slightly) for whiteboard note recognition [3] 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.

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

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

In addition, certain on-line features describing the relation of the sample point $\mathbf{s}_t$ to its neighbors were adopted and altered from those described in [3]. These are:

f_9 : logarithmic transformation of the aspect of the trajectory between the points $\mathbf{s}_{t-\tau}$ and $\mathbf{s}_t$ (referred to as "vicinity aspect"),

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

$f_{10,11}$: angle φ between the line $[\mathbf{s}_{t-\tau}, \mathbf{s}_t]$ and the lower line (coded as $\sin\varphi$ and $\cos\varphi$, the "vicinity slope")
f_{12} : length of trajectory normalized by $\max(|\Delta x|; |\Delta y|)$ ("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} : 3×3 subsampled bitmap slid along pen's trajectory ("context map") to incorporate a 30×30 fraction of the currently written letter's actual image
f_{23-24} : number of pixels above, or respectively, beneath the current sample point $\mathbf{s}_t$ (the "ascenders" and "descenders")

3 Vector Quantization and Discrete HMMs

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

3.1 Vector Quantization

Quantization is the mapping of a continuous, N-dimensional sequence $\mathbf{O} = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$, $\mathbf{f}_t \in \mathbb{R}^N$ to a discrete, one dimensional sequence of codebook indices $\hat{\mathbf{o}} = (\hat{f}_1, \ldots, \hat{f}_T)$, $\hat{f}_t \in \mathbb{N}$ provided by a codebook $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_{N_{\text{cdb}}})$, $\mathbf{c}_k \in \mathbb{R}^N$ containing $|\mathbf{C}| = N_{\text{cdb}}$ centroids $\mathbf{c}_i$ [8]. 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

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

where $d(\mathbf{f}_t, \mathbf{c}_k)$ is commonly the squared Euclidean distance. The codebook $\mathbf{C}$ itself and its entries 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 described in e. g. [8; 9; 10]. As stated in [8], the centroids of a well trained codebook capture the distribution of the underlying feature vectors $p(\mathbf{f})$ in the training data.

As the values of the features described in Sec. 2.1 are neither mean nor variance normalized, each feature f_j is normalized to the mean $\mu_j = 0$ and standard derivation $\sigma_j = 1$, yielding the normalized feature $\tilde{f}_j$. The statistical dependencies of the features are thereby unchanged.

3.2 Discrete HMMs

For handwriting recognition with discrete HMMs each symbol (in case of 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 [1]. In order to use discrete HMMs, the continuous observations $\mathbf{O} = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ are vector quantized yielding discrete observation sequences $\mathbf{o}_i = (\hat{f}_1, \ldots, \hat{f}_T)$ as explained in the previous section. Given some discrete training data $\mathbf{o}_i = (\hat{f}_1, \ldots, \hat{f}_T)$ the parameters λ_i can be trained with the well known EM-method, in the case of HMMs known as Baum-Welch-algorithm [11]. Recognition is performed by presenting the unknown pattern $\mathbf{x}$ to all HMMs λ_i and selecting the model

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

with the highest likelihood. In case of word or even sentence recognition this is done by the Viterbi algorithm [12], which also performs a segmentation of the input vector $\mathbf{x}$.

4 VQ Designs

As mentioned in the introduction, in contrast to ASR, both continuous and discrete features are used in handwritten whiteboard note recognition. In case of discrete HMMs, where all features are quantized jointly, the binary feature f_1 as defined in Eq. 1 looses its significance, due to inadequate quantization and quantization error. However, in [6] it is stated that in continuous HMM based recognition systems the pen's pressure is one of the four most significant features. In the following, we therefore present two VQ design which explicitly model the binary pressure feature f_1 without loss.

4.1 Joint-Codebook VQ Design

For the first VQ design, $\tilde{f}_1$ is treated to be statistically independent of f_{2-23}[2], i. e. $p(\tilde{f}_1|\tilde{f}_2, \ldots, \tilde{f}_{24}) = p(\tilde{f}_1)$. In this case

$$p(\tilde{\mathbf{f}}) = p(\tilde{f}_1, \ldots, \tilde{f}_{24}) = p(\tilde{f}_1) \cdot p(\tilde{f}_2, \ldots, \tilde{f}_{24}). \tag{4}$$

Hence, $p(\tilde{\mathbf{f}})$ can be quantized by two independent codebooks. The first one is directly formed by the normalized pressure value $\tilde{f}_1$, and the remaining features f_{2-23} are represented by M 23-dimensional centroids $\mathbf{r}_k$. N centroids $\mathbf{c}_i$ representing the 24-dimensional data vectors $\tilde{\mathbf{f}}_t = (\tilde{\mathbf{f}}_1, \ldots, \tilde{f}_{24})$ can then be obtained by augmenting the centroids $\mathbf{r}_k$ as follows:

$$\mathbf{c}_i = \begin{cases} (\tilde{f}_1|_{f_1=0}\,,\, \mathbf{r}_j) & 1 \le i \le M, \qquad\qquad j = i \\ (\tilde{f}_1|_{f_1=1}\,,\, \mathbf{r}_j) & M+1 \le i \le 2 \cdot M,\; j = i - M, \end{cases} \tag{5}$$

where $\tilde{f}_1|_{f_1=n}$ is the normalized value of $\tilde{f}_1$ corresponding to $f_1 = n$, $n = \{0, 1\}$. Hence, Eq. 5 describes the construction of a joint codebook. The total number N of centroids $\mathbf{c}_i$ calculates to $N = 2 \cdot M$. The modified VQ system is shown in Fig. 1.

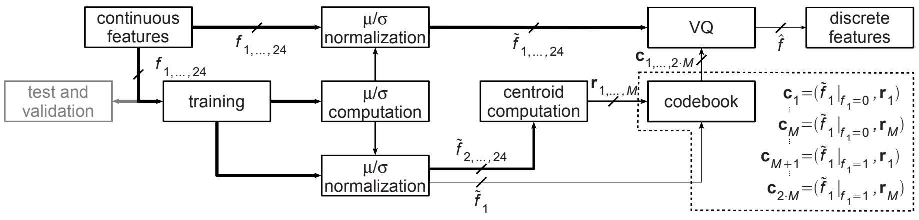
continuous features
$f_{1,\ldots,24}$
μ/σ normalization
$\tilde{f}_{1,\ldots,24}$
VQ
$\hat{f}$
discrete features
$f_{1,\ldots,24}$
$\mathbf{c}_{1,\ldots,2\cdot M}$
test and validation
training
μ/σ computation
centroid computation
$\mathbf{r}_{1,\ldots,M}$
codebook
$\mathbf{c}_1 = (\tilde{f}_1|_{f_1=0}, \mathbf{r}_1)$
$\mathbf{c}_M = (\tilde{f}_1|_{f_1=0}, \mathbf{r}_M)$
$\mathbf{c}_{M+1} = (\tilde{f}_1|_{f_1=1}, \mathbf{r}_1)$
$\mathbf{c}_{2\cdot M} = (\tilde{f}_1|_{f_1=1}, \mathbf{r}_M)$
$\tilde{f}_{2,\ldots,24}$
μ/σ normalization
$\tilde{f}_1$

Fig. 1. VQ system using a joint codebook for explicitly modeling f_1 and assuming statistical independence according to Eq. 4.

[2] Note: the authors are aware that in this case also multiple-stream HMMs [4] can be applied. However, these are out of this paper's scope.

4.2 Codebook-Switching VQ Design

In the second VQ design the statistical dependency between f_1 and the other features is taken into account. Applying Bayes' rule the joint probability $p(\tilde{\mathbf{f}})$ in this case yields

$$\begin{aligned} p(\tilde{\mathbf{f}}) &= p(\tilde{f}_1, \ldots, \tilde{f}_{24}) = p(\tilde{f}_2, \ldots, \tilde{f}_{24} | \tilde{f}_1) \cdot p(\tilde{f}_1) = \\ &= \begin{cases} p(\tilde{f}_2, \ldots, \tilde{f}_{24} | \tilde{f}_1 < 0) \cdot p(\tilde{f}_1 < 0) & \text{if } f_1 = 0 \\ p(\tilde{f}_2, \ldots, \tilde{f}_{24} | \tilde{f}_1 > 0) \cdot p(\tilde{f}_1 > 0) & \text{if } f_1 = 1. \end{cases} \end{aligned} \tag{6}$$

As pointed out in Sec. 2.1 the feature f_1 is binary. Hence, as indicated in Eq. 6 $p(\tilde{\mathbf{f}})$ can be represented by two arbitrary codebooks $\mathbf{C}_\mathrm{s}$ and $\mathbf{C}_\mathrm{g}$ depending on the value of f_1.

To adequately model $p(\tilde{f}_2, \ldots, \tilde{f}_{24} | \tilde{f}_1)$, the normalized training set $\tilde{\mathcal{S}}_\mathrm{train}$, which consists of T_train feature vectors $(\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_{T_\mathrm{train}})$, is divided into two sets $\mathcal{F}_\mathrm{s}$ and $\mathcal{F}_\mathrm{g}$ where

$$\begin{array}{l} \mathcal{F}_\mathrm{s} = \{\tilde{\mathbf{f}}_t | f_{1,t} = 0\} \\ \mathcal{F}_\mathrm{g} = \{\tilde{\mathbf{f}}_t | f_{1,t} = 1\} \end{array}, \ 1 \leq t \leq T_\mathrm{train}. \tag{7}$$

Afterwards, the assigned feature vectors are reduced to the features $f_{2,\ldots,24}$, yet the pressure information can be inferred from the assignment of Eq. 7. N_s centroids $\mathbf{r}_{\mathrm{s},i}$, $i = 1, \ldots, N_\mathrm{s}$ are derived from the set $\mathcal{F}_\mathrm{s}$ and N_g centroids $\mathbf{r}_{\mathrm{g},j}$, $j = 1, \ldots, N_\mathrm{g}$ from set $\mathcal{F}_\mathrm{g}$ forming two *independent* codebooks $\mathbf{R}_\mathrm{s}$ and $\mathbf{R}_\mathrm{g}$ holding $N = N_\mathrm{s} + N_\mathrm{g}$ centroids for the whole system. Given N and the ratio

$$R = \frac{N_\mathrm{g}}{N_\mathrm{s}} \Rightarrow N_\mathrm{g} = \left\lfloor \frac{N}{1 + \frac{1}{R}} + 0.5 \right\rfloor, N_\mathrm{s} = N - N_\mathrm{g} \tag{8}$$

the number of N_s and N_g can be derived for any value of N. The optimal value of both, N and R with respect to maximum recognition accuracy is derived by experiment in Sec. 5.

In order to keep the exact pressure information after VQ for each normalized feature vector $\tilde{\mathbf{f}}_t$ of the data set, first two codebooks $\mathbf{C}_\mathrm{s}$ and $\mathbf{C}_\mathrm{g}$ (with centroids $\mathbf{c}_{\mathrm{s},i}$, $1 \leq i \leq N_\mathrm{s}$ and $\mathbf{c}_{\mathrm{g},j}$, $1 \leq j \leq N_\mathrm{g}$) are constructed similar to Eq. 5:

$$\begin{array}{ll} \mathbf{c}_{\mathrm{s},i} = (\tilde{f}_1|_{f_1=0}\,, \mathbf{r}_{\mathrm{s},i}) & , 1 \leq i \leq N_\mathrm{s} \\ \mathbf{c}_{\mathrm{g},j} = (\tilde{f}_1|_{f_1=0}\,, \mathbf{r}_{\mathrm{g},j}) & , 1 \leq j \leq N_\mathrm{g}. \end{array} \tag{9}$$

The value of the feature $\tilde{f}_{1,t}$ is handed to the quantizer separately to decide which codebook $\mathbf{C}$ should be used for quantization:

$$\hat{f}_t = \begin{cases} \underset{1 \leq i \leq N_\mathrm{s}}{\operatorname{argmin}}\, d(\tilde{\mathbf{f}}_t, \mathbf{c}_{\mathrm{s},i}) & \text{if } f_{1,t} = 0 \\ \left[\underset{1 \leq j \leq N_\mathrm{g}}{\operatorname{argmin}}\, d(\tilde{\mathbf{f}}_t, \mathbf{c}_{\mathrm{g},j}) \right] + N_\mathrm{s} & \text{if } f_{1,t} = 1. \end{cases} \tag{10}$$

The VQ system using two switching codebooks is illustrated in Fig. 2.

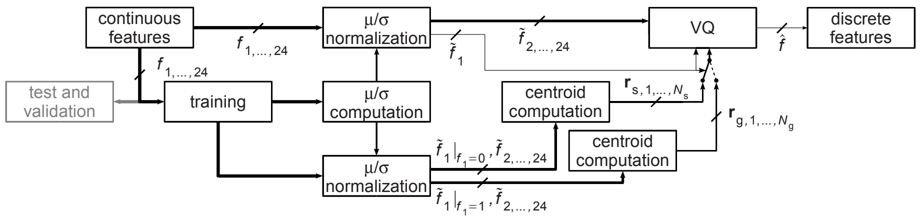

Fig. 2. Second VQ system to model the statistical dependency between f_1 and $f_{2,\ldots,24}$ according to Eq. 6

5 Experimental Results

The experiments presented in this section are conducted on a database containing handwritten heuristically line-segmented whiteboard notes (IAM-OnDB[3]). For further information on the IAM-OnDB see [13]. Comparability of the results is provided by using the settings of the writer-independent IAM-onDB-t1 benchmark, consisting of 56 different characters and an 11 k dictionary which also provides well-defined writer-disjunct sets (one for training, two for validation, and one for test). For our experiments, the same HMM topology as in [3] is used.

The following four 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 feature vectors ($f_{1,\ldots,24}$) 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 $a_{\text{base}} = 62.6\,\%$ 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 [1].

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, all features except the pressure information ($f_{2,\ldots,24}$) are quantized jointly for the second experiment. As Fig. 3 (left) shows, only little degradation in recognition performance compared to the baseline can be observed. The peak rate of $a_{\text{r}} = 62.5\,\%$ is once again reached at a codebook size of $N = 5000$, which equals a relative change of $r = -0.2\,\%$. This is rather surprising as in [6] pressure is assumed to be a relevant feature in on-line whiteboard note recognition.

Experiment 3. (*Exp. 3*): The fact the recognition accuracy decays only moderately when omitting the pressure information shows that the binary information is not quantized properly in the first experiment. Therefore f_1 is directly

[3] `http://www.iam.unibe.ch/~ fki/iamnodb/`

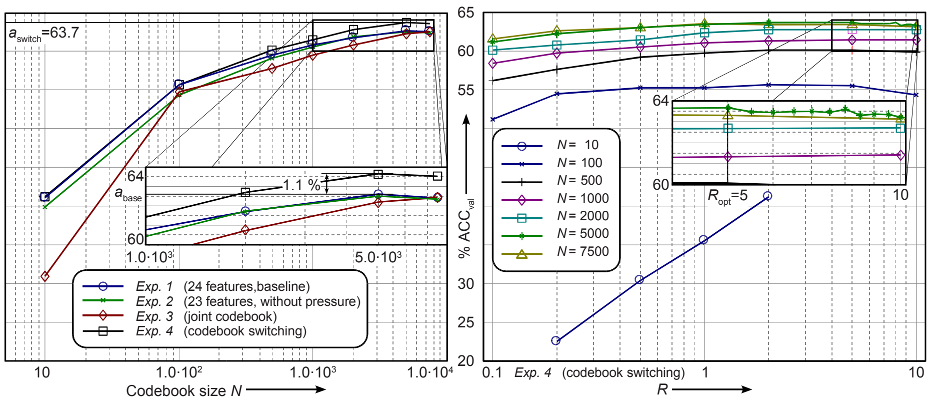

Fig. 3. Evaluation of different systems' character accuracy with respect to the codebook size N (left), character accuracy for different codebook sizes and varying ratios $R = \frac{N_g}{N_s}$ for the VQ design using codebook-switching (right)

modeled as explained in Sec. 4.1 and a joint codebook is used. This results in a further drop in performance due to the independent modeling of f_1 and f_{2-24} as displayed in Fig. 3 (left). Peak performance is found to be $a_{\text{joint}} = 62.4$ for a codebook size of $N = 7500$ which is a relative change of $r = -0.3\,\%$.

Experiment 4. (*Exp. 4*): In the last experiment the performance of the second VQ system, as introduced in Sec. 4.2, is evaluated. The optimal value of $R = \frac{N_g}{N_k}$ is found by experimentation. 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 formerly found optimal values of R. The highest character accuracy of $a_{\text{switch}} = 63.7\,\%$ is found for $N = 5000$ and $R_{\text{opt}} = 5$, which yields (according to Eq. 8) $N_s = 833$ 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 $r = 1.8\,\%$ ($\varDelta R = 1\,\%$ absolute).

In order to prove the competitiveness of our systems, the parameters and models which delivered the best-performing systems in the previous experiments are used 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 [14]. The baseline system, using a standard VQ and coding all features jointly achieves, a word accuracy of $A_{\text{base}} = 63.5\,\%$. As expected from the character accuracy of the previous experiments (*Exp. 2*), the omission of the "pressure" information has little influence on the word-level accuracy: $A_{\text{r}} = 63.3\,\%$ can be reported in this case, indicating a drop of $r = -0.3\,\%$ relative to the baseline system. The word accuracy of the first VQ design using a joint codebook is $A_{\text{joint}} = 63.2\,\%$, a relative change of $r = -0.5\,\%$ given the baseline. Compared to the standard VQ system an absolute word accuracy of $A_{\text{switch}} = 65.6\,\%$ can be achieved by using the codebook-switching design (*Exp. 4*) which is a relative improvement of $r = 3.3\,\%$. This system even slightly outperforms the system presented in [14] by $r = 0.6\,\%$ relative.

Table 1. Final results for the experiments *Exp1*, ..., *Exp4* and one different system [14] on the same word-level recognition task

system	*Exp. 1* (A_{base})	*Exp. 2* (A_{r})	*Exp. 3* (A_{joint})	*Exp. 4* (A_{switch})	[14]
word-level accuracy	63.5 %	63.3 %	63.2 %	65.6 %	65.2 %

6 Conclusion and Discussion

In this paper, we successfully applied discrete HMMs in the field of handwritten whiteboard note recognition. Our experiments with a common VQ system show that the binary pressure information is not adequately quantized regardless of the codebook size. To overcome this problem, two VQ designs are introduced which model the presure information without loss. The first approach, based on the assumption that the pressure feature is statistically independent of the remaining features, showed improvement neither in character nor in word accuracy. The second VQ design takes the statistical dependency between the pressure and the remaining features into account by using two arbitrary codebooks. The main parameters for this second system are the ratio $R = \frac{N_{\text{g}}}{N_{\text{s}}}$ of the two codebook sizes as well as the total number N of codebooks used. Both parameters are optimized on a validation set by means of maximal 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 $r = 0.6\,\%$ can be reported, illustrating the competitiveness of our system.

In future work the role of the parameter R will be investigated more analytically. Additionally, in the future 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 [4].

Acknowledgments

The authors sincerely thank Marcus Liwicki for providing the lattice for the final benchmark, Garrett Weinberg, and the reviewers for their useful comments.

References

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

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

[3] Liwicki, M., Bunke, H.: HMM-Based On-Line Recognition of Handwritten Whiteboard Notes. In: Proc. of the Int. Workshop on Frontiers in Handwriting Rec., pp. 595–599 (2006)
[4] Rigoll, G., Kosmala, A., Rottland, J., Neukirchen, C.: A Comparison between Continuous and Discrete Density Hidden Markov Models for Cursive Handwriting Recognition. In: Proc. of the Int. Conf. on Pattern Rec., vol. 2, pp. 205–209 (1996)
[5] Schenk, J., Rigoll, G.: Novel Hybrid NN/HMM Modelling Techniques for On-Line Handwriting Recognition. In: Proc. of the Int. Workshop on Frontiers in Handwriting Rec., pp. 619–623 (2006)
[6] Liwicki, M., Bunke, H.: Feature Selection for On-Line Handwriting Recognition of Whiteboard Notes. In: Proc. of the Conf. of the Graphonomics Society, pp. 101–105 (2007)
[7] Kavallieratou, E., Fakotakis, N., Kokkinakis, G.: New Algorithms for Skewing Correction and Slant Removal on Word-Level. In: Proc. of the Int. Conf. ECS, vol. 2, pp. 1159–1162 (1999)
[8] Makhoul, J., Roucos, S., Gish, H.: Vector Quantization in Speech Coding. Proc. of the IEEE 73(11), 1551–1588 (1985)
[9] Forgy, E.W.: Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications. Biometrics 21, 768–769 (1965)
[10] Gray, R.M.: Vector Quantization. IEEE ASSP Magazine, 4–29 (April 1984)
[11] Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions of Finite State Markov Chains. Annals of Mathematical Statistics 37, 1554–1563 (1966)
[12] Viterbi, A.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Transactions on Information Theory 13, 260–267 (1967)
[13] Liwicki, M., Bunke, H.: IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard. In: Proc. of the Int. Conf. on Document Analysis and Rec., vol. 2, pp. 1159–1162 (2005)
[14] Liwicki, M., Bunke, H.: Combining On-Line and Off-Line Systems for Handwriting Recognition. In: Proc. of the Int. Conf. on Document Analysis and Rec., pp. 372–376 (2007)