# Discrete Single Vs. Multiple Stream HMMs: A Comparative Evaluation of Their Use in On-Line Handwriting Recognition of Whiteboard Notes

*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 paper we study the influence of quantization on the features used in on-line handwriting recognition in order to apply discrete single and multiple stream HMMs. It turns out that the separation of the features in statisticaly independent streams influences the performance of the recognition system: using the discrete "pressure" feature as an example, we show that the statistical dependencies between the features are as important as their proper quantization.*

*In an experimental section we show that a continuous state-of-the-art system for on-line handwritten whiteboard note recognition can be outperformed by $r = 2.0\,\%$ relative in word level accuracy using a three stream discrete HMM system with well chosen streams. A relative improvement of $r = 4.5\,\%$ can be achieved when comparing a single stream HMM system with the best performing mutliple stream HMM system.*

**Keywords:** Discrete HMM, discrete multiple stream HMM, on-line whiteboard note recognition, quantization

## 1. Introduction

In modern on-line handwriting recognition, Hidden-Markov-Models (HMMs, [15; 18]) have proven to be the classifier of choice [13], due to their capability of modeling time-dynamic sequences of variable lengths. HMMs also compensate for statistical variations in those sequences. More recently, they were introduced for on-line handwritten whiteboard note recognition [9].

Commonly, in a handwriting recognition system, each symbol (for the purposes of this paper, each character) is represented by a single HMM (either discrete or continuous). By combining character-HMMs, words are recognized using a dictionary. While high recognition rates are reported for *isolated word* recognition systems, e. g. [5], performance considerably drops when it comes to unconstrained handwritten *sentence* recognition [9]: 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 [9]. Due to the conditions described in [9], one may characterize the problem of on-line whiteboard note recognition as "difficult".

One distinguishes between continuous and discrete HMMs. The latter are further divided into discrete single and multiple stream HMMs [16]. In case of continuous HMMs, the observation probability is modeled by mixtures of Gaussians [15]. It is known that the performance of a handwriting recognition systems depends on both the number of Gaussians used and the number of iterations for which each Gaussian is trained [4]. Therefore, exhaustive optimizations are needed when using continuous HMMs. In case of discrete HMMs, the probabilities are estimated by counting each discrete symbol's occurrences [15]. The probability density function of the observation is thereby derived exactly and not approximated by Gaussians, as in case of continuous HMMs. However, in order to transform the continuous handwriting data into discrete symbols, vector quantization (VQ) is performed, which introduces a quantization error. While in automatic speech recognition (ASR) continuous HMMs have gained wide acceptance, it remains unclear whether discrete or continuous HMMs should be used in on-line handwriting [16] and whiteboard note recognition in particular.

In this paper we discuss the use of discrete single and multiple stream HMMs for the task of on-line whiteboard note recognition with respect to varying

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

codebook-sizes and the number of streams. While ASR features generally are continuous in nature, in handwriting recognition continuous, discrete, and binary features are used [9]. We prove that the feature describing the pen "pressure" is not quantized adequately when performing vector quantization due to quantization errors. In this paper we therefore form groups of features which are quantized separately and show that an improvement can be achieved using certain multiple stream configurations.

To that end, the next section gives a brief overview of the general preprocessing and feature extraction system for whiteboard note recognition. Section 3 reviews VQ and both discrete single and multiple stream HMMs. Five different discrete HMM recognition systems used and evaluated in this paper are briefly summarized in Sec. 4. In an experimental section (Sec. 5), the database used for evaluation is explained and the formerly introduced systems are evaluated. In addition, our system is compared to a state-of-the-art continuous system to prove competitiveness. Finally, conclusions and an outlook for further work are presented in Sec. 6.

## 2. System Overview

For recording the handwritten whiteboard data the eBeam-System[1] is used. For further information refer to [9]. After recording, the sampled data (which is sampled equidistantly neither in time nor space) is preprocessed and normalized as a first step. The data is thereby resampled in order to achieve an equidistant sampling in space. Then, a histogram-based skew- and slant-correction is performed as described in [7]. Finally all text lines are normalized to meet a distance of "one" between the upper and lower baseline similar to [3].

Following the preprocessing, 24 features are extracted from the recorded data, resulting in a 24-dimensional feature vector $\mathbf{f}(t) = (f_1(t), \ldots, f_{24}(t))^{\mathrm{T}}$ derived from the three-dimensional data vector (sample points) $\mathbf{s}_t = (x(t), y(t), p(t))^{\mathrm{T}}$. The state-of-the-art features for handwriting recognition [6] and recently published new features for whiteboard note recognition [9] used in this paper are briefly listed below. Commonly, the features 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. $f_1 = 1$ if the pen's touches hits the whiteboard and $f_1 = 0$ otherwise.

$f_2$ : 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 $\alpha$ of spatially resampled and normalized strokes (coded as sin $\alpha$ and cos $\alpha$, and called "writing direction")

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

In addition, we use on-line features which describe the relation between a sample point $\mathbf{s}_t$ and its neighbors as described in [9] and altered.

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

$$v = \left(\frac{\Delta y - \Delta x}{\Delta y + \Delta x}\right), \qquad \begin{array}{l}\Delta x = x(t) - x(t-\tau) \\ \Delta y = y(t) - y(t-\tau),\end{array}$$

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

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

$f_{10,11}$ : angle $\varphi$ between the line $[\mathbf{s}_{t-\tau}, \mathbf{s}_t]$ and lower line (coded as sin $\varphi$ and cos $\varphi$, and called "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 off-line features are:

$f_{14-22}$ : a $3 \times 3$ subsampled bitmap slid along pen's trajectory ("context map") to incorporate a $30 \times 30$ partition of the currently written letter's actual image

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

## 3. Discrete Single and Multiple Stream HMMs and Vector Quantization

In this section we briefly summarize discrete Hidden Markov Models (HMMs) and discrete multiple stream HMMs from a Graphical Model's (GM, [2]) point of view and give a common notation. Then vector quantization (VQ) is reviewed and the necessary feature normalization for on-line whiteboard note recognition is explained.

### 3.1. Discrete single stream HMMs

The GM of a discrete single stream HMM $\lambda$ with the variable parameters $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ and hidden states $s_1, \ldots, s_N$ is depicted in Fig. 1 left, where $q_t$ denotes the state $s_i$ which is occupied at time instance $t$. Matrix $\mathbf{A}$, consisting of the entries $a_{ij} =$

[1] http://www.ebeam.com

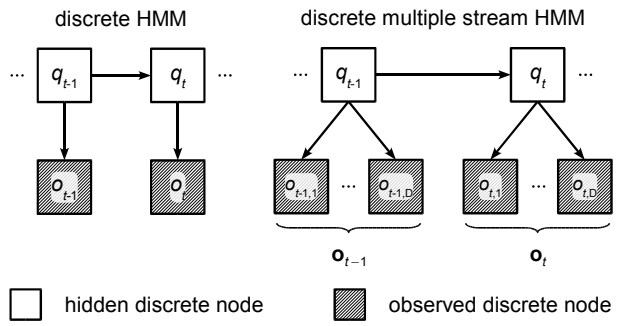


**Figure 1**. GM of a discrete single stream HMM (left) and a discrete multiple stream HMM (right).

$p(q_t = s_j|q_{t-1} = s_i)$ thereby describes the time-invariant probability of a state transition $q_{t-1} \rightarrow q_t$, $\mathbf{B}$, with entries $b_{s_i}(o_t) = p(o_t|s_i)$ the discrete emission-probability of each state $s_i$ for each possible symbol $o_t$ and $\pi = (\pi_i, \ldots, \pi_N)$ the initial state distribution $\pi_i = p(q_1 = s_i)$ [15]. Given a certain parameter set $\lambda$, the discrete HMM's joint probability of the observation $\mathbf{o} = (o_1, \ldots o_T)$ and the state sequence $\mathbf{q} = (q_1, \ldots, q_T)$ can be calculated to

$$p(\mathbf{o}, \mathbf{q}|\lambda) = p(q_1) \cdot p(o_1|q_1) \cdot \prod_{t=2}^{T} p(q_t|q_{t-1}) \cdot p(o_t|q_t). \quad (1)$$

By marginalizing, that is summing up Eq. 1 over all possible state sequences $\mathbf{q} \in \mathbf{Q}$, and using the above substitutions $a_{ij}$, $b_{s_i}(o_k)$ and $\pi_i$, the well-known production probability

$$p(\mathbf{o}|\lambda) = \sum_{q \in \mathbf{Q}} \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} b_{q_t}(o_t) \quad (2)$$

is derived and can be computed efficiently using the forward-algorithm [15]. The parameters $\lambda$ of a HMM can be trained using EM-Algorithm, in case of HMMs known as Baum-Welch-algorithm [1].

### 3.2. Discrete multiple stream HMMs

In GM-notation, a discrete multiple stream HMM with $D$ observation streams and parameters $\lambda_{\mathrm{St}} = (\mathbf{A}, \mathcal{B}, \pi)$ producing the observation sequence $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]$ is shown on the right hand side of Fig. 1. It is the discrete counterpart of the "Multi-Stream" HMM with equally weighted streams, as e. g. described in [14]. While the parameters $\mathbf{A}$ and $\pi$ have the same definition as for single stream HMMs, the production probabilities of the discrete multiple stream HMM are stored separately for each stream $d$ in $\mathcal{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_D\}$. At each time instance $t$ the observation $\mathbf{o}_t = [o_t(1), \ldots, o_t(D)]$ is produced, where the observation $o_t(d)$ of stream $d$ is produced *statistically independently* from all other streams given the current state $q_t$. This is indicated in Fig. 1:

$$p(\mathbf{o}_t|s_i) = \prod_{d=1}^{D} p(o_t(d)|s_i) = \prod_{d=1}^{D} b_i(o_t(d)) \equiv l_i(\mathbf{o}_t). \quad (3)$$

The joint probability of the observation sequence $\mathbf{O}$ and the state sequence $\mathbf{q} = (q_1, \ldots, q_T)$ is then, similarly to Eq. 1, given by

$$p(\mathbf{O}, \mathbf{q}|\lambda_{\mathrm{St}}) = p(q_1) \cdot \prod_{d=1}^{D} p(o_1(d)|s_i) \cdot \prod_{t=2}^{T} p(q_t|q_{t-1}) \cdot \prod_{d=1}^{D} p(o_t(d)|s_i). \quad (4)$$

Again, by marginalizing and using the substitutions $a_{ij}$, $\pi_i$, and Eq. 3 the production probability yields

$$p(\mathbf{O}|\lambda) = \sum_{q \in \mathbf{Q}} \pi_{q_1} l_{q_1}(\mathbf{o}_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} l_{q_t}(\mathbf{o}_t). \quad (5)$$

As indicated by the similar structure of Eqs. 2 and 5, the training algorithms for the discrete HMM as described in [1] have to be slightly modified to handle multiple streams.

### 3.3. Vector Quantization

In order to use discrete single or multiple stream HMMs, all continuous observations $\mathbf{O}$ are assigned to a stream of discrete observations $\mathbf{o}$ via quantization, whereby continuous, $N$-dimensional sequence $\mathbf{O} = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$, $\mathbf{f}_t \in \mathbb{R}^N$ of length $T$ is mapped 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 $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_{N_{\mathrm{cdb}}})$, $\mathbf{c}_k \in \mathbb{R}^N$ containing $|\mathbf{C}| = N_{\mathrm{cdb}}$ centroids $\mathbf{c}_i$ [12]. 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

$$\hat{f}_t = \underset{1 \leq k \leq N_{\mathrm{Cdb}}}{\operatorname{argmin}} \; d(\mathbf{f}_t, \mathbf{c}_k), \quad (6)$$

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 $\mathcal{S}_{\mathrm{train}}$ containing $|\mathcal{S}_{\mathrm{train}}| = N_{\mathrm{train}}$ training samples $\mathbf{O}_i$ by partitioning the $N$-dimensional feature space defined by $\mathcal{S}_{\mathrm{train}}$ into

$N_{\text{cdb}}$ cells. This is performed by the well known $k$-Means algorithm as described e. g. in [12]. As stated in [12], 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 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 $\tilde{f}_j$. Thereby the statistical dependencies of the features are not changed.

## 4. Discrete HMM systems

In this section we briefly summarize the five different system topologies that are evaluated in Sec. 5.

**System 1**: The first system comprises the baseline system in which all normalized features $(\tilde{f}_1, \ldots, \tilde{f}_{24})$ are jointly quantized by one codebook and mapped to the discrete symbol $\hat{f}$. This results in the single observation stream $\mathbf{o} = (\hat{f}_1, \ldots, \hat{f}_T)$. As only one observation stream exists recognition is performed by single stream HMMs as described in Sec. 3.1. The same baseline system has been used in our previous work [17].

**System 2**: To prove that the binary "pressure" feature $f_1$ is not adequately quantized by standard VQ, the second system is used. All features *except* $f_1$ are jointly quantized by the discrete symbol $\hat{f}_{\text{r}}$, resulting in one single observation stream $\mathbf{o} = (\hat{f}_{\text{r},1}, \ldots, \hat{f}_{\text{r},T})$. Again, the single stream observations are recognized by single stream HMMs (Sec. 3.1) (see also [17]).

**System 3**: In order to keep the exact value of feature $f_1$, its values are directly used to form an independent observation, where $o_t(1)$ in Eq. 3 becomes $o_t(1) = f_1(t)$, using the unnormalized values of the feature. The remaining features $(\tilde{f}_2, \ldots, \tilde{f}_{24})$ are jointly quantized as in the second system, forming a second observation $o_t(2) = \hat{f}_{\text{r}}(t)$. The two-stream observation $\mathbf{o}_t = (f_1(t), \hat{f}_{\text{r},t})$ is then recognized using two-stream HMMs as explained in Sec. 3.2.

**System 4**: Besides the discrete "pressure," we use other discrete features: the off-line features $f_{14-24}$. In this system the values of each discrete feature, including $f_1$, form a separate stream. The remaining features $(f_{2-13})$ are jointly quantized and mapped by the discrete symbol $\hat{f}_{\text{on}}$. Thus a 13-stream observation $\mathbf{o}_t = (f_1, \hat{f}_{\text{on}}, f_{14}, \ldots, f_{24})$ is derived and a 13-stream HMM is applied for recognition.

**System 5**: The last system is motivated by e. g. [16], in which the on-line and off-line features form two separate observation streams. In this paper we augment this approach by a further discrete observation stream formed by the values of $f_1$, whereby the on-line features $\tilde{f}_{2-13}$ are jointly quantized and mapped to the discrete symbol $\hat{f}_{\text{on}}$ and the off-line features $\tilde{f}_{14-24}$ are mapped to the discrete symbol $\hat{f}_{\text{off}}$. Hence, a three-stream observation $\mathbf{o}_t = (f_1, \hat{f}_{\text{on}}, \hat{f}_{\text{off}})$ is obtained. As the on-line and off-line features are quantized independently their codebook sizes $N_{\text{on}}$ and $N_{\text{off}}$ may differ. The optimal ratio $R = {}^{N_{\text{off}}}/_{N_{\text{on}}}$ is found by experiment.

## 5. Experiments

The experiments presented in this section are conducted on a database containing handwritten, heuristically line-segmented whiteboard notes (IAM-OnDB[2], [8]). Comparability of the results is provided by using the settings of the writer-independent IAM-onDB-t1 benchmark. The IAM-onDB-t1 benchmark provides four writer-disjunct sets of arbitrary size: one training set, two validation sets and a test set. In this paper the training set is used for training, the *combination* of both validation sets is used for validation, and the final tests are performed on the test set. Our experiments use the same HMM topology and number of states as in [9].

The following five experiments are conducted on the combination of both validation sets and on each system described in Sec. 4 at different codebook sizes. As the number of observation streams varies, the number $N$ of "feature describing parameters" is fixed, i. e. $N = \sum_{d=1}^{D} N_d$ where $N_d$ is the number of codebook entries of each observation stream $d$. To achieve comparability all experiments are performed for values of $N = 10$, 100, 500, 1000, 2000, 5000 and 7500.

**Experiment 1** (*Exp. 1*): Using the first system, this experiment forms the baseline: all features are quantized in one single stream. The results derived on the validation sets are shown in Fig. 2 left. The best character level accuracy is achieved for $N_{\text{cdb}} = 5000$ prototypes and yields $a_{\text{b}} = 62.6\,\%$. The drop in performance when further increasing the codebook size to $N = 7500$ is due to sparse data [15].

**Experiment 2** (*Exp. 2*): The inadequate quantization of the binary "pressure" feature is proven by experiments on the second system. As Fig. 2 left shows, only 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 $a_{\text{sys}\,2} = 62.5\,\%$ is once again reached for a codebook size of $N_{\text{cdb}} = 5000$, which equals a relative change of $r = 0.2\,\%$. This is rather surprising as in [11] pressure is proven to be a relevant feature in on-line whiteboard note recognition.

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

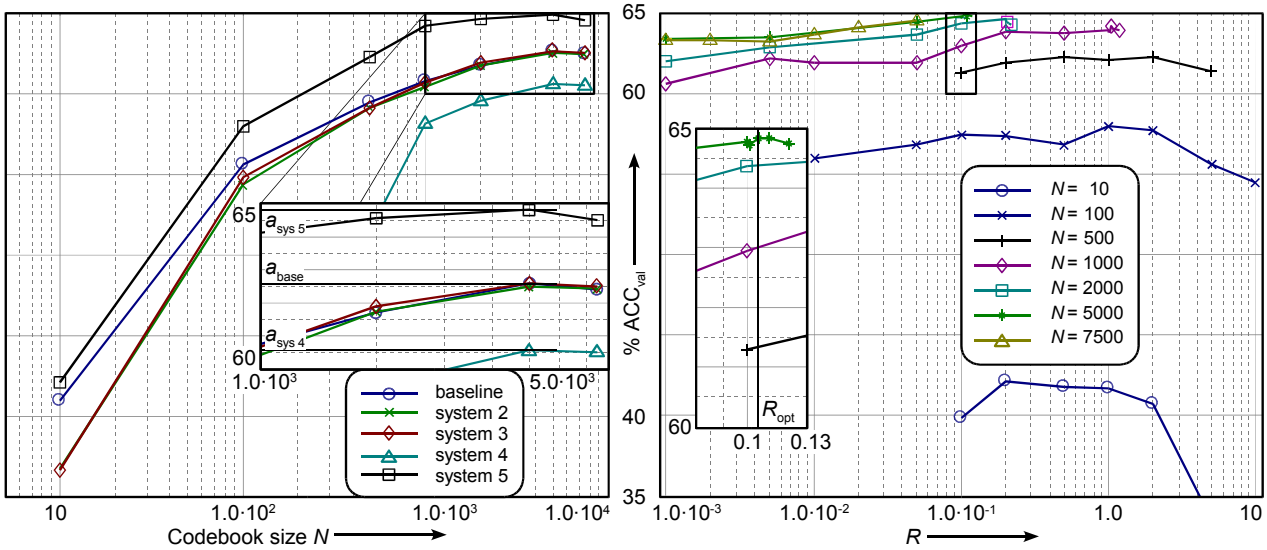


**Figure 2**. Left: evaluation of different systems' character level accuracies with respect to the codebook size $N$. Right: character level accuracies for different codebook sizes and varying ratios $R = {}^{N_{\text{off}}}/_{N_{\text{on}}}$ of a three-stream HMM system (where the three streams are formed by the values of the pressure, the jointly quantized on-line features using $N_{\text{on}}$ centroids, and the jointly quantized off-line features using $N_{\text{off}}$ centroids).

**Experiment 3** (*Exp. 3*): The result of the previous experiment suggests that the discrete pressure information of feature $f_1$ is not adequately quantized by the vector quantizer. The third system allows the direct use of the values of $f_1$ as a separate observation stream. The pressure information is thereby modeled without loss. This results in no improvement. As mentioned in Sec. 3.2, both observation streams are modeled statistically independently. Although modeled without loss in a separate stream, the neglect of the statistical dependencies between the pressure and the remaining features inhibits any improvement.

**Experiment 4** (*Exp. 4*): The previous experiment's outcome might show that neglecting the statistical dependencies between the features influences the recognition accuracy. Because in on-line handwritten whiteboard note recognition other discrete features besides the "pressure" feature are used, they can be modeled in separate observation streams without loss using the fourth system. The exact values of each feature are thereby modeled without any loss while their statistical dependencies are neglected. Peak performance of $a_{\text{sys}\,4} = 60.6\,\%$ is reached for $N = 5000$, which is a relative drop of $r = -3.3\,\%$ compared to the baseline system. This drop confirms the assumption that the statistical dependencies between features have an influence on the system's performance.

**Table 1**. Final results of experiments 1,...,5 and a state-of-the-art continuous system [10] on the same word level recognition task.

| system | *Exp 1* ($A_{\text{b}}$) | *Exp 2* ($A_{\text{sys}\,2}$) | *Exp 3* ($A_{\text{sys}\,3}$) | *Exp 4* ($A_{\text{sys}\,4}$) | *Exp 5* ($A_{\text{sys}\,5}$) | [10] |
|---|---|---|---|---|---|---|
| word acc. | 63.5 % | 63.3 % | 63.6 % | 61.0 % | 66.5 % | 65.2 % |

**Experiment 5** (*Exp. 5*): In this last experiment the fifth system is evaluated. Three independent observation streams are formed consisting of the values of feature $f_1$, the jointly quantized on-line features $f_{2-13}$, and the jointly quantized off-line features $f_{14-24}$. Both parameters, $N = 2 + N_{\text{off}} + N_{\text{on}}$ and the ratio $R = {}^{N_{\text{off}}}/_{N_{\text{on}}}$ (where $N_{\text{on}}$ and $N_{\text{off}}$ denotes the number of centroids used to quantize the on-line and off-line features), are found by experiment. Fig. 2 shows at right the character level accuracy at given ratio $R$ for different $N$. The peak rates which are achieved for each $N$ and corresponding optimal $R$ are plotted in Fig. 2 left. The highest character level accuracy rate of $a_{\text{sys}\,5} = 64.8\,\%$ is reached for $N = 5000$ and $R_{\text{opt}} = 0.105$, i. e. $N_{\text{on}} = 4523$ and $N_{\text{off}} = 475$ centroids are used to quantize the on-line and off-line features respectively. This is a relative improvement of $r = 3.4\,\%$ compared to the baseline.

In order to prove the competitiveness of our systems the parameters and models which provide the best results in the experiments *Exp. 1*,..., 5 are taken to perform word level recognition on the test set of the IAM-onDB-t1 benchmark. The results are compared with the performance of a state-of-the-art continuous recognition system [10] and shown in Tab. 1. The highest word level accuracy of $A_{\text{sys}\,5} = 66.5\,\%$ can be reported for the $5^{\text{th}}$ system using three independent feature streams leading to an improvement of $r = 2.0\,\%$ relative ($R = 1.3\,\%$ absolute) as compared to the state-of-the-art system. A straightforward approach, i. e. jointly quantizing all features in a single stream, leads to a word level accuracy of $A_{\text{b}} = 63.5\,\%$. The three stream approach as perfomed in system five therefore leads to a relative improvement of $r = 4.5\,\%$ compared to this baseline system.

## 6. Conclusions and Outlook

In this paper we intensively investigated the use of discrete single and multiple stream HMMs. We examined both the continuous and discrete features typically used in on-line handwritten whiteboard note recognition and their vector quantizations. These quantizations introduce quantization errors. By conducting a series of experiments we showed, using the "pressure" feature as example, that both the quantization of the features and the statistical dependencies between them influence recognition performance. This assumption is confirmed by the observation that recognition accuracy considerably drops if the statistical dependencies between the features are neglected, even though they are modeled without loss in separate streams. However, we also showed that a single-stream discrete HMM baseline system using jointly quantized features can be outperformed by $r = 4.5\,\%$ relative in word level accuracy using a three-stream discrete HMM system which models the pressure, the on-line features and the off-line features independently.

Finally our discrete systems were compared to a state-of-the-art continuous system as presented in [10], yielding an improvement of $r = 2.0\,\%$ relative in word level accuracy.

As our experiments indicate, the statistical dependencies between the features have an impact on the system's performance. In future work we therefore plan to study these influences in further detail in order to achieve an optimal feature stream selection.

### Acknowledgments

The authors thank M. Liwicki for providing the lattice for the final benchmark and G. Weinberg for his 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] J. Bilms, "Graphical Models and Automatic Speech Recognition", M. Johnson, S. Khudanpur, M. Ostendorf and R. Rosenfeld, editors, *in: Mathematical Foundations of Speech and Language Processing*, pp 191–246. Springer, 2004.

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

[4] 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.

[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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] J. Makhoul, S. Roucos and H. Gish, "Vector Quantization in Speech Coding", *Proc. of the IEEE*, 73(11):1551–1588, November 1985.

[13] 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.

[14] A. P.S. and K. A.K., "Automatic Facial Expression Recognition Using Facial Animation Parameters and Multistream HMMs", *IEEE Transactions on Information Forensics and Security*, pp 1–9, 2006.

[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", *Proc. of the 30th Symposium of DAGM*, pp 229–238, 2008.

[18] 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.