GRAPHICAL MODELS: STATISTICAL INFERENCE VS. DETERMINATION

Joachim Schenk and Benedikt Hörnler and Artur Braun and Gerhard Rigoll

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

ABSTRACT

Using discrete Hidden-Markov-Models (HMMs) for recognition requires the quantization of the continuous feature vectors. In handwritten whiteboard note recognition it turns out that the pen-pressure information, which is important for recognition, is not adequately quantized and looses significance. In this paper, the implicit modeling of the pressure information presented in previous work which uses the deterministic knowledge on the actual pressure is generalized using a Graphical Model (GM) representation based on statistical inference. The results of two state-of-the-art toolboxes implementing HMMs and GMs are compared. It can be seen that the statistical inference approach based on GMs is inferior to the implicit modeling of the pressure information. It is shown that a direct implementation of HMMs outperforms the mathematic identical GM representation.

***Index Terms*—** GMs, handwriting recognition, VQ

1. INTRODUCTION

Hidden-Markov-Models (HMMs, see [1]) have proven their power for modeling time-dynamic sequences of variable lengths. HMMs also compensate statistical variations in those sequences. Due to this property they have become quite popular in automatic speech recognition (ASR). The field of ASR and on-line handwriting recognition (HWR) are closely related: using a speech recognizer based on HMMs for on-line HWR has been introduced in [2] for the first time — on the ICASSP 1986. Since then, the use of HMMs in on-line HWR has been deeply investigated.

In a common HWR system, each symbol (i. e. character) is represented by one HMM. Words are recognized by combining character-HMMs using a dictionary. While high recognition rates are reported for *isolated word* recognition systems [3], performance considerably drops when it comes to recognition of whole unconstrained handwritten *text lines* [4]: the lack of previous word segmentation introduces new variability and therefore requires more sophisticated character recognizers. An even more demanding task is the HWR of whiteboard notes as introduced in [4], which plays an important role in so-called "smart meeting room" scenarios (see e. g. [5]): when writing on a whiteboard, the writer stands rather than sits and the writing arm does not rest. Therefore, additional variation is introduced. It also has been observed that size and width of characters and words vary on a higher degree on whiteboards than on tablets. These conditions contribute to the characterization of the problem of on-line whiteboard note recognition as "difficult".

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, see [6]) is performed to transform the continuous data to discrete symbols.

In [7] we showed that the important pen pressure information (the importance of this feature has been proven in [8]) gets lost when VQ is performed. This observation has been confirmed for a number of different VQ approaches in [9]. In [7; 9], we introduced a novel VQ scheme incapable of modeling the pressure information without loss and respecting the statistical dependencies between the features.

Graphical Models (GMs, see [10]) are a natural enhancement of HMMs. Hereby following [10], GMs are a combination of probability theory and graph theory, providing a visual graphical language and efficient algorithms for probability calculations and decision making. In this paper, we use the GM notation to model the statistical dependencies as hypothesized in [7; 9]. In contrast to our previous work, in case of GM the statistical dependencies between the pressure information and the remaining features are derived from the data rather by statistical inference than by deterministic knowledge. In a series of experiments, we show that the statistical inference approach is inferior to our previously published approach where the pressure information is modeled implicitly by deterministic knowledge. In addition, in this paper we compare two toolboxes for building HMM-based and GM-based recognition systems, either of handwriting or speech recognition, namely the "Hidden-Markov-Toolkit" [11] and the "Graphical Models Toolkit" [12]. While the experiments and the used features are native to HWR, the toolboxes are not. Hence, the general results may be of interest even for the signal processing and speech community.

The next section gives an overview of the HWR system for whiteboard notes, reviews VQ, and introduces discrete HMMs in a GM notation. The statistical bindings between the pressure information and the remaining features are hypothesized and expressed in GM notation in Sec. 3. An evaluation and comparison of the proposed model to previous results is given in the experimental section (Sec. 4). Finally, conclusion and discussion are presented in Sec. 5.

2. SYSTEM OVERVIEW

In this section, we summarize our HWR system. Then a short review on VQ and discrete HMMs is given.

2.1. Preprocessing and Feature extraction

The x- and y-coordinates as well as the pen's "pressure" p of the handwritten, heuristically line-segmented whiteboard notes are recorded using the EBEAM-System as explained in [4]. Hence, the handwritten script is described by the sample vectors $\mathbf{s}(t) = (x(t), y(t), p(t))^{\mathsf{T}}$. Afterwards a resampling of the data is

performed followed by a correction of the skew and the slant of the script trajectory, using a histogram-based approach as explained in [13]. Finally, all text lines are normalized to meet a distance of "one" between the corpus and the base line.

Following the preprocessing, 24 state-of-the-art *on-line* and *off-line* features are extracted. The extracted on-line features are: the pen's "pressure", indicating whether the pen touches the whiteboard surface (f_1); a velocity equivalent, which is computed before re-sampling (f_2); the x- and y-coordinate after resampling ($f_{3,4}$); the "writing direction", i. e. the angle α of the strokes, coded as $\sin\alpha$ and $\cos\alpha$ ($f_{5,6}$); and the "curvature", i. e. the difference of consecutive angles $\Delta\alpha = \alpha(t) - \alpha(t-1)$, coded as $\sin\Delta\alpha$ and $\cos\Delta\alpha$ ($f_{7,8}$); a logarithmic transformation of the "vicinity aspect" v, $\text{sign}(v) \cdot \log(1+|v|)$ (f_9); 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 $\cos\varphi$ ($f_{10,11}$); and the "vicinity curliness", the length of the trajectory normalized by $\max(|\Delta x|;|\Delta y|)$ (f_{12}). Finally the average square distance to each point in the trajectory and the line $[\mathbf{s}(t-\tau), \mathbf{s}(t)]$ is given (f_{13}).

The off-line features are: a 3×3 "context map" to incorporate a 30×30 partition of the currently written letter's image (f_{14-22}); and "ascenders" and "descenders" (i. e. the number of pixels above respectively beneath the current sample point) ($f_{23,24}$).

2.2. Vector Quantization

Quantization is the mapping of a continuous, D-dimensional sequence $\mathbf{O} = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$, $\mathbf{f}_t \in \mathbb{R}^D$ 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 $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_{N_{\text{cdb}}})$, $\mathbf{c}_k \in \mathbb{R}^D$ containing $|\mathbf{C}| = N_{\text{cdb}}$ centroids $\mathbf{c}_i$ [6]. For $D = 1$ this mapping is called *scalar*, and in all other cases ($D \geq 2$) *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

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

where $d(\mathbf{f}_t, \mathbf{c}_k)$ is commonly the squared Euclidean distance, i. e.

$$d(\mathbf{f}_t, \mathbf{c}_k) = (\mathbf{f}_t - \mathbf{c}_k)^{\text{T}} \cdot (\mathbf{f}_t - \mathbf{c}_k). \tag{2}$$

The codebook $\mathbf{C}$ and its entries c_i are derived from a training set $\mathcal{S}_{\text{t}}$ containing $|\mathcal{S}_{\text{t}}| = N_{\text{t}}$ training samples $\mathbf{O}_i$ by partitioning the D-dimensional feature space defined by $\mathcal{S}_{\text{t}}$ into N_{cdb} cells. This is performed by the k-Means algorithm as described in e. g. [6]. As stated in [6], the centroids of a 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 vector $\tilde{\mathbf{f}}_t = (\tilde{f}_{1,t}, \ldots, \tilde{f}_{D,t})$.

2.3. Discrete Hidden-Markov-Model

The Graphical Model (GM, see [10]) of a discrete HMM λ with the variable parameters $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ and hidden states $s_1, \ldots, s_N$ is shown in Fig. 1 where q_t denotes the state s_i which is occupied at time instance t. The matrix $\mathbf{A}$ consisting of the entries $a_{ij} = p(q_t = s_j | q_{t-1} = s_i)$, 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 state s_i for the symbol o_t, and $\pi = (\pi_i, \ldots, \pi_N)$ the initial state distribution $\pi_i = p(q_1 = s_i)$ [1]. Given

Fig. 1. GM of a discrete HMM.

a certain parameter set λ, the joint probability of the observation $\mathbf{o} = (o_1, \ldots, o_T)$ and the state sequence $\mathbf{q} = (q_1, \ldots, q_T)$ yields

$$p(\mathbf{o}, \mathbf{q}|\lambda) = p(q_1) \cdot p(o_1|q_1) \cdot \prod_{t=2}^{T} p(q_t|q_{t-1}) \cdot p(o_t|q_t). \tag{3}$$

By marginalizing, i. e. summing Eq. 3 over all possible state sequences $\mathbf{q} \in \mathbf{Q}$, and using the above substitutions a_{ij}, $b_{s_i}(o_t)$ and π_i, the production probability,

$$p(\mathbf{o}|\lambda) = \sum_{q \in \mathbf{Q}} \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1} q_t} b_{q_t}(o_t), \tag{4}$$

is derived, and can be computed efficiently using the forward-algorithm [1]. The parameters λ of a HMM are trained using the Baum-Welch-algorithm [14]. Combined recognition and segmentation is enabled by the Viterbi-algorithm [1].

3. VARIANTS OF PRESSURE MODELING

In this section, we address the problem of adequately modeling the pressure information: as pointed out in [8], the pressure is an important feature in on-line HWR of whiteboard notes. However, in [7; 9] we showed that this feature looses its significance during quantization. Hence, review the method of modeling the pressure without loss while respecting its statistical dependencies to the remaining features, the switching codebook, and give two possible GM representations.

3.1. Switching Codebook

As pointed out in [7; 9], the statistical bindings between the pressure information and the remaining features are important when VQ is performed. Applying Bayes' rule, the joint probability $p(\tilde{\mathbf{f}}) = p(\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_{24})$ of the features can be written as

$$\begin{aligned} p(\tilde{\mathbf{f}}) &= p(\tilde{f}_1, \ldots, \tilde{f}_{24}) = p(\tilde{f}_2, \ldots, \tilde{f}_{24}|\tilde{f}_1) \cdot p(\tilde{f}_1) = \\ &= \begin{cases} p(\tilde{f}_2, \ldots, \tilde{f}_{24}|\tilde{f}_1 < 0) \cdot p(\tilde{f}_1 < 0) & \text{if } f_1 = 0 \\ p(\tilde{f}_2, \ldots, \tilde{f}_{24}|\tilde{f}_1 > 0) \cdot p(\tilde{f}_1 > 0) & \text{if } f_1 = 1. \end{cases} \end{aligned} \tag{5}$$

As pointed out in Sec. 2 the feature f_1 is binary. Hence, as indicated in Eq. 5, $p(\tilde{\mathbf{f}})$ can be represented by two arbitrary codebooks $\mathbf{C}_{\text{s}}$ and $\mathbf{C}_{\text{g}}$ 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 $\tilde{\mathcal{S}}_{\text{t}}$, which consists of T_{t} feature vectors $(\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_{T_{\text{t}}})$, is divided into two sets $\mathcal{F}_{\text{s}}$ and $\mathcal{F}_{\text{g}}$ where

$$\mathcal{F}_{\text{s}} = \{\tilde{\mathbf{f}}_t | f_{1,t} = 0\},\ \mathcal{F}_{\text{g}} = \{\tilde{\mathbf{f}}_t | f_{1,t} = 1\},\ 1 \leq t \leq T_{\text{t}}. \tag{6}$$

As this separation is done depending on the actual value of the pressure feature, the modeling of the pressure is deterministic. Then, the assigned feature vectors are reduced by f_1, yet the pressure information can be inferred from the assignment of Eq. 6. N_{s} centroids $\mathbf{r}_{\text{s},i}$, $i = 1, \ldots, N_{\text{s}}$ are derived from the set $\mathcal{F}_{\text{s}}$ and N_{g} centroids $\mathbf{r}_{\text{g},j}$, $j = 1, \ldots, N_{\text{g}}$ from set $\mathcal{F}_{\text{g}}$ forming two *independent* codebooks $\mathbf{R}_{\text{s}}$

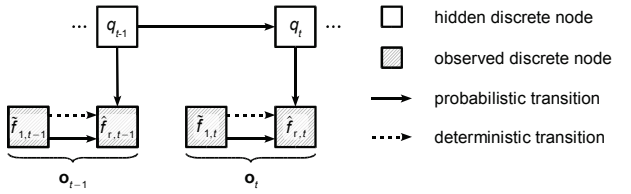

Fig. 2. GM for modeling the pressure feature without loss and respecting its statistical relation to the remaining features by deterministic knowledge (⇢) and statistical inference (→).

and $\mathbf{R}_\mathrm{g}$ holding $N_\mathrm{cdb} = N_\mathrm{s} + N_\mathrm{g}$ centroids for the whole system. N_s and N_g may differ and their values are chosen according to [7].

In order to keep the exact pressure information after VQ for each normalized feature vector $\tilde{\mathbf{f}}_t$ of the data set, the value of the feature $\tilde{f}_{1,t}$ is handed to the quantizer. This deterministic knowledge enables the quantizer to choose the proper codebook $\mathbf{C}$ for quantization.

3.2. Graphical Model representation

The modeling of the pressure feature as described in Eqs. 5 and 6 using GM notation is shown in Fig. 2, accounting for the dashed arrows (⇢): the statistical dependency between the pressure information and the remaining features is modeled by a switching codebook which is determined by the actual value of the pressure feature. Hence, deterministic knowledge is used for modeling the pressure information.

Modeling the features depending on the pressure information is also deductible from the GM shown in Fig. 2. The statistical dependency between the pressure information and the remaining features can be gained by statistical inference [10]. Therefore, the reduced feature vector $\tilde{\mathbf{f}}_\mathrm{r} = (\tilde{f}_2, \ldots, \tilde{f}_{24})$ is quantized to the symbol $\hat{f}_\mathrm{r}$. The statistical dependency between the pressure information and the remaining features is *learned* during training. This is expressed by exchanging the dashed arrows (⇢) by solid arrows (→) in Fig. 2. In contrast to the previous approach, no deterministic switching is performed. The joint probability of the observation $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ and the state sequence $\mathbf{q} = (q_1, \ldots, q_T)$ is derived to

$$p(\mathbf{O}, \mathbf{q}|\lambda) = p(q_1)p(\hat{f}_{\mathrm{r},1}|q_1, \tilde{f}_{1,1})p(\tilde{f}_{1,1}) \cdot \cdot \prod_{t=2}^{T} p(q_t|q_{t-1})p(\hat{f}_{\mathrm{r},t}|q_t, \tilde{f}_{1,t})p(\tilde{f}_{1,t}) \quad (7)$$

and, again by marginalizing and using the abbreviations as introduced in 2.3, the joint probability $p(\mathbf{O}|\lambda)$ yields

$$p(\mathbf{O}|\lambda) = \sum_{q \in \mathbf{Q}} \pi_{q_1} \underbrace{p(\hat{f}_{\mathrm{r},1}|\tilde{f}_{1,1})p(\tilde{f}_{1,1})}_{(**)} \cdot \cdot \prod_{t=2}^{T} a_{q_{t-1}q_t} \underbrace{p(\hat{f}_{\mathrm{r},t}|\tilde{f}_{1,t})p(\tilde{f}_{1,t})}_{(**)} . \quad (8)$$

The marking $(**)$ in Eq. 8 shows the joint modeling of the statistical dependencies between the pressure $\tilde{f}_{1,t}$ and the jointly quantized remaining features: the value of the pressure, directly influences the remaining features. Parameter estimation in the GMs is performed by the Junction Tree (JT) algorithm [15]. As for HMMs, recognition and segmentation is performed using the Viterbi algorithm.

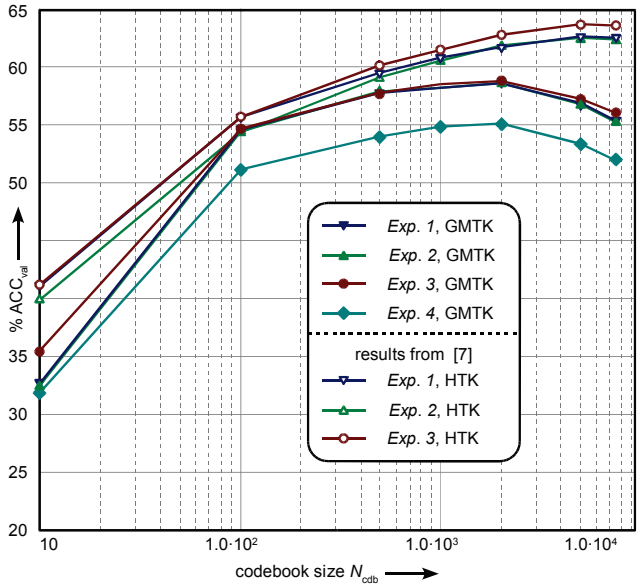

Fig. 3. Evaluation of different systems' character accuracy with respect to the codebook size N_cdb.

4. EXPERIMENTAL RESULTS

Our experiments are conducted on the IAM-onDB-t1 benchmark of the IAM-OnDB, a database containing handwritten whiteboard notes. For further information on the IAM-OnDB see [16]. The writer-independent IAM-onDB-t1 benchmark consists of 56 different characters, and provides writer-disjunct sets (one for training, two for validation, and one for testing). For our experiments, the same HMM topology as in [4] is used. While the experiments in our previous work [7; 9] are conducted using the Hidden-Markov-Toolkit (HTK, see [11]), the experiments in this paper are performed with the Graphical Models Toolkit (GMTK, see [12]), realizing statistical inference [15].

The following four experiments are conducted on the combination of both validation sets, each with seven different codebook sizes (N_cdb = 10, 100, 500, 1 000, 2 000, 5 000, 7 500). For training the vector quantizer, the parameters λ_i of the discrete HMMs, and the parameters of the GM, the IAM-onDB-t1 training set is used. The results with respect to the actual codebook size N_cdb are depicted as *character accuracies* (ACC) in Fig. 3. The first three experiments are a repetition of the experiments presented in [7; 9], while the last experiment evaluates the GM depicted in Fig. 2 (solid arrows).

Experiment 1 (*Exp. 1*): In the first experiment all components of the feature vectors $(\tilde{f}_1, \ldots, \tilde{f}_{24})$ are quantized jointly by one codebook. The results shown in Fig. 3 form the baseline for the following experiments. The maximum character ACC of $a_\mathrm{b} = 58.7\,\%$ is achieved for a codebook size of $N_\mathrm{cdb} = 2\,000$. The drop in recognition performance when raising the codebook size to $N_\mathrm{cdb} = 5\,000$ and $N_\mathrm{cdb} = 7\,500$ is due to sparse data [1].

Experiment 2 (*Exp. 2*): To prove that the binary pressure feature $\tilde{f}_1$ is not adequately quantized by standard VQ all features except the pressure information $(\tilde{f}_2, \ldots, \tilde{f}_{24})$ are quantized jointly for the second experiment. As Fig. 3 shows, only little degradation in recognition performance compared to the baseline can be observed. The peak rate of $a_\mathrm{r} = 58.7\,\%$ is again reached at a codebook size of $N_\mathrm{cdb} = 2000$.

system	*Exp. 1*	*Exp. 2*	*Exp. 3*	*Exp. 4*
GMTK	58.7 %	58.7 %	58.9 %	55.2 %
HTK	62.6 %	62.5 %	63.7 %	—
Δr	-6.6 %	-6.5 %	-8.1 %	—

Table 1. Summary of the results of the experiments *Exp1*,. . ., *Exp4* and the results presented in [7; 9].

Experiment 3 (*Exp. 3*): In order to model the pressure information without loss and respecting the statistical dependencies to the remaining features, the switching codebook approach as explained in [7; 9] and summarized in GM notation in Fig. 2 (dashed arrows) is used in this experiment. The result is a slight improvement of $\Delta r = 0.3\,\%$, and a peak character ACC of $a_{\mathrm{m1}} = 58.9\,\%$ is achieved. This confirms the findings of our previous work.

Experiment 4 (*Exp. 4*): While in the previous experiment the pressure information is modeled by deterministic knowledge, i. e. the pressure information determines which codebook is used for quantization, in this experiment the GM shown in Fig. 2 (solid arrows) is evaluated: the statistical bindings between the pressure information and the remaining features are found by statistical inference during training. Therefore the quantized, reduced feature vector of *Exp. 2* is used and combined with the pressure information. The results are shown in Fig. 3. The peak character ACC of $a_{\mathrm{m2}} = 55.2\,\%$ is reached for $N_{\mathrm{cdb}} = 2000$ which denotes a relative change of $6.3\,\%$. This reveals the superiority of the utilization of the deterministic knowledge as proposed in [7; 9].

The result of the experiments *Exp. 1*, . . . ,*Exp. 4* and the results presented in [7] are summarized in Tab. 1. When comparing these results, a relative reduction of the character ACC of $\Delta r = -6.6\,\%$, $\Delta r = -6.5\,\%$, and $\Delta r = -8.1\,\%$, respectively can be observed. The direct implementation of the HMMs and the codebook switching design performs significantly better then the more general but less optimized GM approach.

5. CONCLUSION AND DISCUSSION

In this paper, the experiments presented in our previous work [7; 9] using discrete HMMs for HWR of whiteboard notes have been repeated, confirming an interesting observation: while in continuous HMM-based HWR of whiteboard notes the pressure information is a vital feature [8], the significance of this feature gets lost when VQ is performed. We therefore reviewed a recently published method for implicitly modeling the pressure information while accounting for the statistical dependencies to the remaining features, utilizing the pressure information as deterministic knowledge. In this paper, this method has been generalized such that the statistical dependencies between the pressure information and the remaining features is learned from the training set.

A series of experiments delivered three major outcomes. First, the observation that the pressure information looses significance if quantized ([7; 9]) is confirmed even when using a GM based toolbox instead of a toolbox optimized for HMMs. Second, a GM-implementation (realized in the GMTK) of the HMM-based HWR system is outperformed by the optimized HMM implementation in the HTK. In addition, it has been shown that the dependency between the pressure information and the remaining features can be estimated by statistical inference. However this approach is clearly outperformed, when the deterministic knowledge about the pressure information is utilized and directly used for recognition. While the recognition systems, features and the distinct outcome of the experiments presented in this paper are native to HWR, the general observation, due to the toolboxes used is not. The results therefore can be of interest even for the speech community.

Although outperformed, the GM representation of our HWR system offers a more flexible and hence, expandable platform for recognition. In future work we therefore plan further investigations of the use of GMs in handwritten whiteboard note recognition. Especially the deterministic binding between the observations in the GMs will be issued in future work.

References

[1] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257 – 285, 1989.

[2] R. Nag, K. Wong, and F. Fallside, "Script Recognition Using Hidden Markov Models," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 11, pp. 2071 – 2074, 1986.

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

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

[5] A. Waibel, T. Schultz, M. Bett, I. Rogina, R. Stiefelhagen, and J. Yang, "SMaRT: The Smart Meeting Room Task at ISL," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 752 – 755, 2003.

[6] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551 – 1588, 1985.

[7] J. Schenk, S. Schwärzler, G. Ruske, and G. Rigoll, "Novel VQ Designs for Discrete HMM On-Line Handwritten Whiteboard Note Recognition," *Proc. 30th Symposium of DAGM*, pp. 234 – 243, 2008.

[8] M. Liwicki and H. Bunke, "Feature Selection for On-Line Handwriting Recognition of Whiteboard Notes," *Proc. Conf. of the Graphonomics Society*, pp. 101 – 105, 2007.

[9] J. Schenk and G. Rigoll, "Neural Net Vector Quantizers for discrete HMM-Based On-Line Handwritten Whiteboard-Note Recognition," *Proc. Int. Conf. on Pattern Recognition*, p. in press, 2008.

[10] J. Bilmes, "Graphical Models and Automatic Speech Recognition," in *Mathematical Foundations of Speech and Language Processing*, M. Johnson, S.P. Khudanpur, M. Ostendorf, and R. Rosenfeld, Eds., pp. 191 – 246. Springer, 2004.

[11] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2.1)*, Cambridge University Engineering Department, 2002.

[12] J. Bilmes and G. Zweig, "The Graphical Model Toolkit: An Open Source Software System for Speech and Time-Series Processing," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 3916 – 3919, 2002.

[13] E. Kuvallieratou, N. Fakotakis, and G. Kokkinakis, "New Algorithms for Skewing Correction and Slant Removal on Word-Level," *Proc. Int. Conf. on Electronics*, vol. 2, pp. 1159 – 1162, 1999.

[14] L.E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Annals of Mathematical Statistics*, vol. 37, pp. 1554 – 1563, 1966.

[15] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Dissertation. University of California, Berkeley, 2002.

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