On Supercomputing in Handwritten Whiteboard Note Recognition

Meetings play an important role in any organizational structure: the Intel Corporation schedules around 3 million meeting hours and another 56 thousand hours each year for training their employees on how to hold a meeting efficiently. High effort is taken to investigate how computers can be used to make meetings more efficient and analyze them automatically (see e.g. [1]), or making expensive meetings available in fully digital form. Therefore, special locations like the IDIAP smart meeting room (see [2]) use cameras, microphones, and a sampling device for recording notes written on a whiteboard. Automatic recognition of such recorded notes is a relatively new and a challenging task in pattern recognition (see [3]) and can be seen as a milestone towards analyzing meetings automatically.

Why using Supercomputers for handwriting Recognition?

Although represented in a similar form as on handheld-devices (so called personal digital assistants, PDAs), which already offer limited capabilities of handwriting recognition, the problem of whiteboard note recognition requires much more sophisticated algorithms, [3]. First, in contrast to handwriting recognition on a PDA, the recognizer cannot be trained on the specific handwriting of a single user: being an instrument to communicate ideas among the participants mutually, writer independent script recognition is required. Second, there is a need to recognize whole text lines rather than isolated words.

Besides, in a whiteboard scenario the writer stands rather than sits and moves the whole arm and body during writing introducing a high level of distortions to the handwriting. In order to test the writer independence of a recognizer, millions of characters have to be validated – a task, which can only be solved in parallel within reasonable time.

Recognition

For recognition of whiteboard notes, up to 24 features are extracted from all sample points of the recorded data, e. g. the pen pressure, describing whether the pen touches the whiteboard, the writing direction and the curvature of the pen trajectory. A complete list and short description can be found in [4]. After the feature extraction, each handwritten character can be described by the corresponding feature sequence. Due to varying writing speeds and writing styles, the same character may consist of a different number of sample points and different feature values. In order to compensate for these dynamic sequences of variable length, it is convenient to represent the character rather by the statistical distribution of its features than by the features themselves. This is realized by so-called Hidden-Markov-Models (HMMs, see [5]) which consist of two statistical processes: one describing the temporal structure of the handwritten character and another for representing the statistics of the observed features. The first process can be described as a Markov-chain, the latter e.g. by mixtures of Gaussian distributions. The estimation of each HMM's parameters, namely the transition probabilities of the underlying Markov-chain and the Gaussian mixtures, is a computational expensive task: starting with just one mixture, the transition probabilities are adjusted from training observations. Afterwards, the number of mixtures is increased by one and the transition probabilities are refined. This continues until a maximum of 32 mixtures is reached.

Recognition is performed by presenting an unknown script pattern to all HMMs and selecting the HMM which yields the highest observation probability for the unknown sample. A maximum-likelihood segmentation and recognition of connected characters forming words and whole text lines can be realized by interconnecting several HMMs. To prevent the models from over-fitting the training data, after each training iteration a validation is necessary. Due to the computationally expensive training procedure and the quite costly recognition process, the validation step after each iteration is usually omitted (see [6]), which compromises the quality of the models. By using the parallel computational capabilities of the HLRB II, a combined training and validation procedure can be realized.

Another goal of our efforts is to use the HLRB II to find a rigorously reduced feature set, achieving an acceptable, writer independent recognition performance and to further use these findings to improve computationally less expensive recognition algorithms (e.g. by replacing the Gaussian mixture distribution of the HMM observation by a discrete distribution) running on low cost hardware. Thus running these algorithms on the HLRB II can be seen as an important step towards robust whiteboard recognition on a standard computer.

Parallelization

In order to provide well-trained models that benefit from validating their parameter after each training iteration the parallelization, which is displayed in Figure 1, is used. Each of the three writer-independent subsets (one for training, one for validation, and one for a final, independent test), containing 5,500,000 training, 3,700,000 validation and 2,900,000 test observation frames, respectively, is partitioned according to the number N of CPUs used for parallelization and forms one instance each for training, validation and testing. For the training, where the model-parameters of the previous iteration are required, so-called

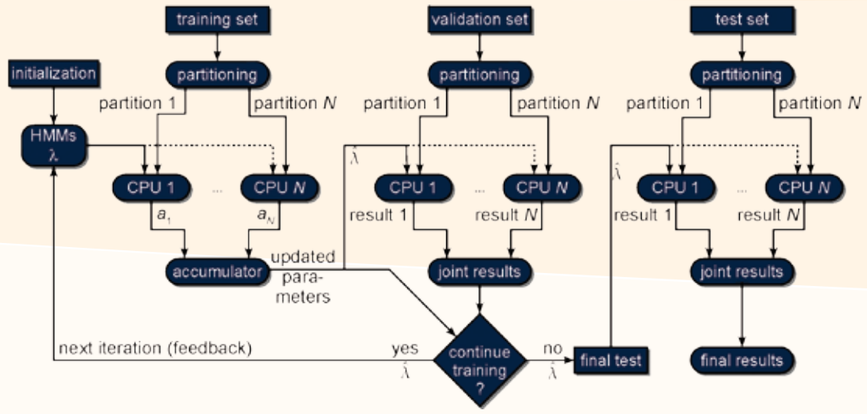

Figure 1: Parallelization scheme for HMM-based recognizer as used in on-line handwriting recognition. The training set is divided into N partitions (test instances) which individually train character models, deriving so-called accumulators (left). In a similar way, the validation (middle) and test procedure (right) can be parallelized.

accumulators "a_i" are the preliminary optimization results. After each training instance has finished calculating its accumulator, all accumulators are combined to a new model. After training, the model parameters are validated. Thereby, each validation instance produces a result, which is again combined to a score for the current model parameters. Depending on the system's design parameters (e.g. the maximum number of iterations or the number of Gaussians), the model parameters are either fed back to the training system providing the new initialization of another training iteration, the number of Gaussian mixtures is increased, or evaluated on the independent test set.

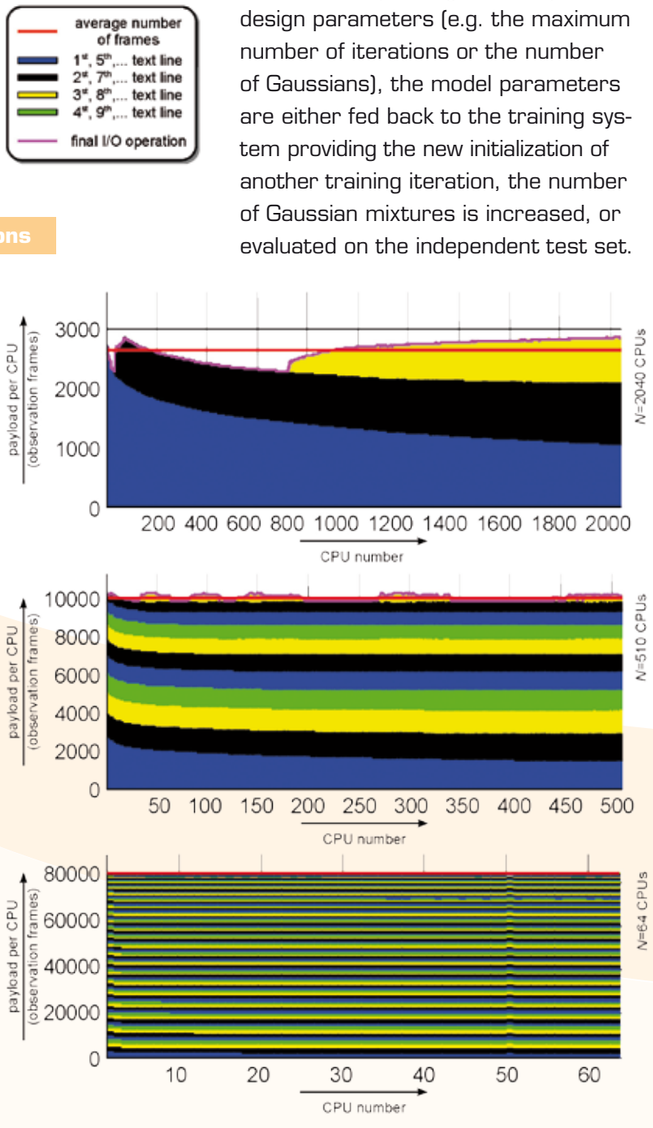

Figure 2: Average (red line) and individual payload per CPU represented by the number of observation frames per CPU and iteration for varying number N of CPUs

A crucial factor influencing the computational performance of the system is the balancing of the training, validation, and test instances. This is provided by a frame-wise partitioning of the data. In order to show the scalability of our parallelization scheme, the number of observation frames processed by each CPU per iteration (the "payload") is shown Figure 2, in case of using either N=64, N=510, or N=2040 CPUs in parallel. Additionally, the average number of observation frames (the average payload) is given. For N=64 and N=510, the CPUs' individual computational payload is well balanced with little deviation from the average payload. In a realistic run, a parallelization using N=64 … 256 CPUs performs adequately efficient. All parameters, results, and data are kept in memory for providing the necessary I/O-operation at the end of each iteration at a high speed.

Preliminary Results

One important task is to lower the computational requirements of a HMM-based handwritten whiteboard note recognizer. Instead of modeling the observation probabilities of the HMMs by mixtures of Gaussians, the features can be quantized by a vector quantizer [4]. Their actual occurrence can then be used as observation probability, leading to discrete HMMs. When performing quantization, it turns out that the pressure information looses its significance (see [4]) as illustrated in Figure 3: the red line denotes the character level accuracy of a_b=62.6% for a discrete HMM system, using all 24 features for quantization. The green line indicates the accuracy of a_r=62.5% of a discrete system where the feature vector has been reduced by the pen's pressure information before quantization – a slight relative drop of r=0.2%.

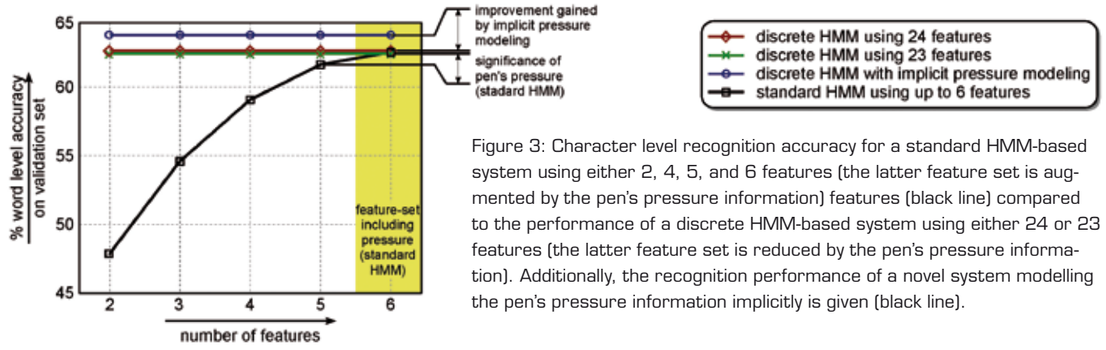

Figure 3: Character level recognition accuracy for a standard HMM-based system using either 2, 4, 5, and 6 features (the latter feature set is augmented by the pen's pressure information) features (black line) compared to the performance of a discrete HMM-based system using either 24 or 23 features (the latter feature set is reduced by the pen's pressure information). Additionally, the recognition performance of a novel system modelling the pen's pressure information implicitly is given (black line).

Additionally, Figure 3 shows the character accuracies of a continuous system using varying numbers of features, chosen by the sequential forward floating selection algorithm [7]. As can be seen, the pressure Information is one of the six most significant features amongst the selected features. These findings motivate a novel VQ-design, enabling an implicit modeling of the pen's pressure information, [4]. The result is also shown: a significant, relative improvement of r=1.8 % can be reported in case of the implicit modeling of the pressure. This shows how the results derived by supercomputing can improve low cost systems.

Outlook

The main reason for using supercomputing in the field of handwritten whiteboard note recognition is to get novel insights: they can enable an improvement of existing recognition algorithms running on devices with less performance. Taking the loss of the information of the pen's pressure, when vector quantization is performed, as an example, investigations conducted on the HLRB II led to an implicit modeling of the pressure information and thereby to a significant improvement. In future work, supercomputers can help to investigate the ability of combining several approaches for handwriting recognition, providing a well-selected set of recognizers and parameters for a more accurate recognition performance at much lower computational cost.

References

[1] **Al-Hames, M., Lenz, C., Reiter, S., Schenk, J., Rigoll, G.**
Robust Multi-Modal Group Action Recognition in Meetings from Disturbed Videos with the Asynchronous Hidden Markov Model, Proceedings of the International Conference on Image Processing, pp. 213–216, 2007

[2] **Moore, D.**
The IDIAP smart meeting room, Research Report 7, 2002

[3] **Liwicki, M., Bunke, H.**
HMM-based on-line Recognition of handwritten Whiteboard Notes, Proceedings of the IWFHR, pp. 595-599, 2006

[4] **Schenk, J., Schwärzler, S., Ruske, G., Rigoll, G.**
Novel VQ Designs for discrete HMM on-line handwritten Whiteboard Note Recognition, Pattern Recognition, LNCS 5096, pp. 234-243, 2008

[5] **Rabiner, L.R.**
A Tutorial on Hidden Markov Models and selected Applications in Speech Recognition, Proceedings of the IEEE, 77(2), pp. 257-285, 1989

[6] **Günter, S., Bunke, H.**
HMM-based handwritten Word Recognition: on the Optimization of the Number of States, Training Iterations, and Gaussian Components, Pattern Recognition, 37(10), pp. 2069-2079, 2004

[7] **Pudil, P., Novovicova, J., Kittler, J.**
Floating Search Methods in Feature Selection, Pattern Recognition Letters, 15(11), pp. 1119-1125, 1994

• Joachim Schenk

Technische Universität München, Departement for Human-Machine Communication

Autumn 2008 • Vol. 6 No. 2 • inSiDE