ISCA Archive
http://www.isca-speech.org/archive

2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

SPEAKER INDEPENDENT CONTINUOUS HMM-BASED RECOGNITION OF ISOLATED WORDS ON A REAL-TIME MULTI-DSP SYSTEM

Abdulmesih Aktas and Klaus Zünkler

Siemens AG, Corporate Research & Development
Otto-Hahn-Ring 6, 8000 München 83, Germany

Abstract

This paper describes a speaker independent continuous Hidden Markov Model recognizer implemented on a real-time multi-DSP system. Training and recognition are based on continuous mixture density HMMs for phonemes. Context dependent triphone models are used and the Viterbi algorithm is applied for both training and recognition. The system is implemented on a workstation with an integrated multi-DSP based acoustic front-end employing three Texas Instruments TMS320C25 signal processors and a Siemens ASIC for vector quantization. In spite of the simplifications made in order to reduce the high computational requirements for the continuous mixture densities, the system has a recognition rate of 99.5% for the speaker independent German digit task with telephone quality.

Keywords: Speech Recognition, Hidden Markov Model, mixture density, speaker independent, real-time.

1. Introduction

Hidden Markov Models (HMM) with continuous mixture densities as stochastic models for speech recognition have been shown to be well-suited for modeling spectral and temporal variabilities in speech [4]. The computation of the emission probabilities required for the probability density functions is usually a highly time consuming task. Therefore most real-time applications of HMM-based recognizers on digital signal processors (DSP) use discrete models (e.g. [3]). For the implementation of the continuous mixture density HMMs powerful floating point DSPs or special ASICs have to be employed. We use in our approach a vector quantization VLSI-chip (VQ) for producing the emission probabilities. For the feature extraction integer DSPs are employed. The Viterbi search is implemented on a RISC-based workstation.

The recognizer was evaluated on two different speaker independent German digit databases, whereby recognition rates of 99.5% were achieved. More details on the recognition experiments are given in [6,7].

The following section of our paper describes the structure of our models. Section 3 deals with the real-time multi-DSP front-end, while section 4 presents implementation aspects of the recognition algorithm on the hardware. The VQ realization of the emission probability computation is explained in more detail. Finally some results conclude the paper.

2. Model Structure

As experiments with word models have shown a poor ability of discriminating confusable words, phoneme models are used in our system [6]. The phonemes are modelled with context dependency because their acoustic realizations are often heavily affected by coarticulation. The HMMs that we use for each phoneme consist of three segments for modeling the beginning, the middle and the end. Each segment consists of two states with tied emission probabilities. The first and the third segment consider coarticulatory effects due to the transitions to the neighboring phonemes, whereas the second segment stands for the middle part which is less affected by the phonetic context.

According to a pronunciation lexicon word models are concatenated of phoneme models. During recognition only phoneme sequences contained in the lexicon are used. On the segment layer triphone models, shown in *Figure 1*, are the basis. In the example the concatenation of three phonemes (/d/, /r/ and /ai/) to the German word '*drei*' is given. The three phonemes are embedded into silence models (/si/). The second line shows the context dependent segments (**si**, ^{si}**d**, **d**, **d**^{r}, ^{d}**r** and so on) with their associated indices, which indicates the adjacent or predecessor segment of the current phoneme for this word. This kind of modeling allows robust and high recognition performance.

On the segment layer the possible number of triphones is reduced because every middle segment of a phoneme model is adjacent only to segments belonging to the same phoneme. So the middle segments of the phoneme models do not have any context information. Each beginning segment of a phoneme is succeeded by the same context independent middle segment. Thus

Amazon / Zentian Limited
Exhibit 1059
Page 1

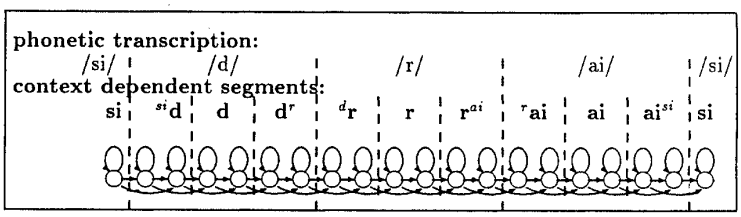

Figure 1: Context dependent phoneme based word model of the German digit '*drei*' (three).

the triphones are reduced into a mixture of bigrams and context independent segments. With an inventory of 40 phonemes approximately 3000 segment models would be possible. In practice, with a small digit vocabulary this is reduced to 90 segment models.

3. Multi-DSP Acoustic Front-End

The recognition system is implemented on a UNIX workstation employing a multi-DSP based acoustic front-end (AkuFE) [1] which was developed within the speaker adaptive continuous speech understanding system SPICOS [4]. In the SPICOS system the AkuFE is employed with the acoustic-phonetic decoding and the real-time recognition of subword units [2]. The AkuFE is designed for real-time speech processing tasks and is a configurable VMEbus system employing Texas Instruments TMS320C25 signal processors and a Siemens ASIC for vector quantization (VQ). A powerful system can be configured using three boards (one master and two slave boards) with a total computational power of more than 100 MIPs. In this case five signal processors and two VQ processors can work in parallel. *Figure 2* shows a general configuration of an extended system.

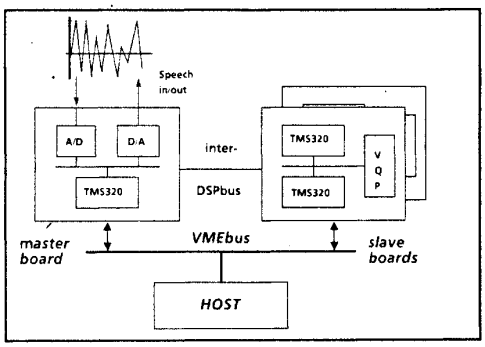

Figure 2: The multi-DSP acoustic front-end system

The master board is the basic control system of an extendable multi-DSP architecture. It consists of an analogue (A/D and D/A converter) and digital, Texas Instruments TMS320C25-based subsystem, whereas the slave board employs two TMS320C25 DSPs and a VLSI-chip for VQ. A local data and program memory is dedicated to each signal processor. A data memory region of 1 KWord is provided on the master board as dual-ported memory enabling a parallel access from both the DSP and the host processor. This allows in case of signal acquisition, a real-time transfer of the preprocessed and analyzed data. The inter-DSPbus allows fast communication between the signal processors on the master and the slave boards without interfering with the VMEbus. A global memory scheme is realized in order to perform data transfers between the processors. For a flexible VMEbus and inter-DSP communication an interrupt control is provided [1].

The VQ processor is able to handle vectors of variable dimensions and codebooks of programmable size. Each codebook vector can be extended up to 64 dimensions of 8 bit components, without noticeable loss of time performance. Four software selectable codebook memory banks each with 1 Mbit are dedicated to the processor. The chip can be controlled from either one of the DSP subsystems of the AkuFE or from the host. The on-chip cache RAM of the VQ processor allows parallel load of a vector during the search. Within 1 second the VQ can compare 10^7 vector components of the codebook to a given vector. Two different distance measures (City-block and Euclid) are supported. The chip delivers the codebook entry and the distance of the best codebook candidate.

4. Real-Time Implementation

Real-Time Considerations

Training and recognition in our system are based on continuous mixture density HMMs for phonemes as used in the SPICOS system [4]. The main difference besides recording conditions and sampling frequencies is that context dependent models are used. The Viterbi algorithm is implemented for both, training and recognition. Our models are left to right first order Markov models with a fixed $\underline{A}$ matrix, N-dimensional feature vector and a continuous mixture density function $p(\underline{x})$ with M modes, which is based on Gaussian density functions :

$$\begin{aligned} p(\underline{x}_t) &= \sum_{m=1}^{M} c_m \mathcal{N}(\underline{x}_{tm}, \vec{\mu}_m, \underline{C}_m) \\ &= \sum_{m=1}^{M} \frac{c_m}{\sqrt{2\pi}} \frac{1}{|\underline{C}_m|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{x}_{tm}-\underline{\mu}_m)\underline{C}_m^{-1}(\underline{x}_{tm}-\underline{\mu}_m)} \quad (1) \end{aligned}$$

where $\underline{x}_t$ represents the input frame vector, c_m is the mixture weight for the m-th mode, $\underline{\mu}_m$ is the mean

Amazon / Zentian Limited
Exhibit 1059
Page 2

vector of the mixture mode m and $\underline{C}_m$ is the covariance matrix.

In order to achieve real-time processing some assumptions concerning the emission probabilities are made [6]:

- The covariance matrix $\underline{C}$ is assumed to be diagonal.
- It is assumed that the contribution of the nearest mixture density is much larger than the contribution of the others, thus the probability is computed only for the best fitting mode k for a given input vector $\underline{x}_t$. Additionally the variances of all mixture densities are considered to be equal.

This results in the reduction of formula (1) to:

$$p_k(\underline{x}_t) = \frac{c_k}{K_1} e^{-K_2 \sum_{n=1}^{N} (x_{tn}-\mu_{kn})^2} \qquad (2)$$

where K_1 and K_2 are constants which depend on the variance of the distributions. Simulation experiments have shown that this is no limitation if there are enough mixtures available for modeling the density functions. With these simplifications the emission probability depends only on the mean of the best fitting mixture density and its mixture weight. In the logarithmic domain, the computation of the probability simplifies to an Euclidian distance

$$-\ln p_k(\underline{x}_t) = K_1 - \ln c_k + K_2 \sum_{n=1}^{N} (x_{tn} - \mu_{kn})^2 \qquad (3)$$

which can be easily computed by the special hardware.

With the assumption of Laplacian density functions the mixture computation can be further reduced to the City-block distance and modified to a distance measure d_{kt} derived from the probabilities:

$$\begin{aligned} d_{kt} &= \frac{-\ln p_k(\underline{x}_t)}{K_2} - \frac{K_1}{K_2} \\ &= \tilde{c}_k + \sum_{n=1}^{N} |x_{tn} - \mu_{kn}|, \quad \text{with } \tilde{c}_k = -\frac{\ln c_k}{K_2} \qquad (4) \end{aligned}$$

$\tilde{c}_k$ is here a mode dependent constant.

Previous experiments with both distance measures have not revealed significant differences in the recognition results.

The Viterbi search algorithm is split into two parts which run on different processors. The production of the emission probabilities for each frame is performed on the VQ processor while the calculation of the accumulated word model probabilities is processed on the host machine. *Figure 3* shows the organization of the system moduls on the hardware.

In the current system three DSPs and one VQ processor are employed for the preprocessing, feature extraction and emission probability calculation. The DSP on the master board and the VQ control processor work in multitasking modes. The first DSP handles

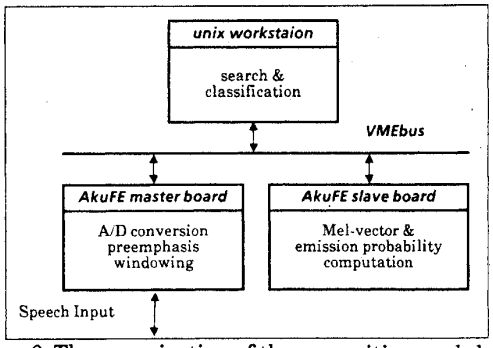

Figure 3: The organization of the recognition moduls on the hardware.

additionally the double-buffered organization of the real-time signal acquisition. The host machine receives for each 10 ms an interrupt which initiates a buffer read on the mailbox of the master board to get the distances for the current frame.

Feature Extraction

Two boards of the AkuFE system are employed for the recognition task. The first board of the AkuFE system performs the preprocessing steps, amplification, low-pass filtering at a corner frequency of 3.4 kHz, sampling with 8 kHz and A/D conversion (12 bit resolution). After the preemphasis a 256-point Hamming window is multiplied to the speech frame with a step size of 10 ms. Further tasks of this board are the control of the slave DSPs and of the data flow over the Mailbox and global memory.

Mel-scaled filter bank energies and their differences are used as feature vector. The time consuming computation of this vector is implemented in integer arithmetic on the slave board employing both of the DSP subsystems. A 256-point FFT is applied to the windowed signal. The spectrum is passed to a filter bank. The Mel-filter bank basically performs a summation of the FFT-channel energies within the selected frequency ranges varying according to the Mel-scale. The filter bank energies are then transformed to the log-domain. This allows a coding of each dimension with 8 bit resolution. In order to eliminate irrelevant energy variations for neighbored frequencies which are caused by the pitch, a cepstral smoothing is applied to the 24-dimensional spectras. The smoothing is done by a convolution of a liftering window transformed into the frequency domain. Additionally the total frame energy, the first and the second derivative of the Mel-vector is computed. After condensation of the derivates the feature vector has 52 dimensions.

Computation of the emission probabilities

The feature vector, which is computed in real-time every 10 ms, is passed to the VQ stage in order to compute the local emission probabilities for the current

Amazon / Zentian Limited
Exhibit 1059
Page 3

frame. The organization of the codebook is as follows: For each segment there are about ten codebook entries containing the mean vector and the mixture weight of each density. The VQ processor performs in real-time a search over all codebook segments. As each segment is assigned to a subword model the vector quantizer chip computes the distance to the mean of the best fitting mixture density. This is based on the simplifications given above in this section (equation (4)). The minimum distance for each frame and segment model is passed to the workstation which evaluates the word-HMMs in parallel and in real-time. As the local emission probabilities are represented in the log-domain, the calculation of the word probabilities is reduced to a summation.

The VQ processor is controlled by one of the slave DSPs. After each search within a segment the VQ generates an interrupt, causing a distance and index read by the DSP.

Viterbi Search

As our system is a phoneme based recognition system each word in the lexicon is built up from a sequence of phoneme units. For the purpose of word boundary detection a silence model is added to the beginning and end of each model. The number of states in the word models defines the search space for each word. This remains static for each vocabulary. In our case of small vocabularies the search space for all words is prepared in advance. For the digit task the maximum search space includes 256 states (sum of all states in the word models). For each search space the word model probability has to be calculated. The computation can be done in a parallel manner by considering the current feature vector in each search space of each word. This requires about 8 bytes memory per state. A 10-MIPs workstation is necessary in order to handle this task in real-time.

The end detection of the utterances is checked automatically during the search. The best word candidate is evaluated at the final pause state of each word model and if a candidate is the winner for at least 300 ms it is selected as the best word.

The multi-DSP system is also well-suited for the training purposes of the Hidden Markov Models because the computation of the indices of the best modes by the VQ processor allows us to reestimate the mean values of the distributions.

5. Results and Conclusion

The recognition experiments for this work are described in [6]. The German digit database we use contains 11 words (0-9 and and an alternative pronounciation for the German digit '*two*'(*zwo*)). The training set consisted of utterances spoken by 15 male and 15 female speaker each with four versions in average. Speech data of another 30 male and female speakers (440 utterances) was used for test experiments. The recording was performed over an ISDN telephone device with a sampling rate of 8 kHz. The recognition rate is 99.5% for the test database.

In the current demonstration system the Mel-vector calculation requires about 20 MIPs. The next generation DSP (e.g. TMS320C50) will allow us to perform the whole Mel-vector computation on one DSP.With a faster FFT, which runs now on the first slave DSP it is possible to reorganize the moduls and extend the number of the segments. That means, new words can be added to the lexicon. Currently the VQ requires 6 ms and about 40 MIPs for the search, given a segment based codebook. A possibility to double the vocabulary is to employ two VQ processor at a time. The time intensive implementation of the feature extraction algorithms is justified under the aspects of transferring the algorithms to VLSI chips. Expensive silicon space can be saved on this way.

As our system is phoneme based the emission probability production can be easily extended to continuous speech recognition applications. For the 1000 word vocabulary of SPICOS around 130 context dependent segments has to be considered in real-time.

References

[1] A. Aktas and H. Höge: "Multi-DSP and VQ-ASIC Based Acoustic Front-End for Real-Time Speech Processing Tasks", Proc. EUROSPEECH '89, Paris, pp.586-589

[2] A. Aktas and H. Höge: "Real-Time Recognition of Subword Units on a Hybrid Multi-DSP/ASIC Based Acoustic Front-End", Proc. of the ICASSP '89, Glasgow, pp.101-103

[3] A. Ciaremella et.al.: "A PC-housed speaker independent large vocabulary continuous telephonic speech recognizer", in these Proceedings

[4] H. Höge: "SPICOS II - A speech understanding dialogue system", Proc. of the Int. Conf. on Spoken Language Processing (ICSLP), Kobe, Japan (1990)

[5] H. Ney and A. Noll: "Phoneme modeling using continuous mixture densities", Proc. of the ICASSP '88, New York, pp.437-440

[6] K. Zünkler: "An ISDN speech server based on speaker independent continuous Hidden Markov Models", Proc. of NATO-ASI, Cetraro, July 1990

[7] K. Zünkler: "A Discriminative Recognizer for Isolated and Continuous Speech Recognition Using Statistical Information Measures", in these Proceedings

Amazon / Zentian Limited
Exhibit 1059
Page 4