Implementing a Simple Continuous Speech Recognition System on an FPGA

S J Melnikoff, S F Quigley & M J Russell

Electronic, Electrical and Computer Engineering, University of Birmingham,
Edgbaston, Birmingham, B15 2TT, United Kingdom
S.J.Melnikoff@iee.org, S.F.Quigley@bham.ac.uk,
M.J.Russell@bham.ac.uk

Abstract

Speech recognition is a computationally demanding task, particularly the stage which uses Viterbi decoding for converting pre-processed speech data into words or sub-word units. We present an FPGA implementations of the decoder based on continuous hidden Markov models (HMMs) representing monophones, and demonstrate that it can process speech 75 times real time, using 45% of the slices of a Xilinx Virtex XCV1000.

1 Introduction

Real time continuous speech recognition is a computationally demanding task, and one which tends to benefit from increasing the available computing resources.

A typical speech recognition system starts with a pre-processing stage, which takes a speech waveform as its input, and extracts from it feature vectors or observations which represent the information required to perform recognition. This stage is efficiently performed by software. The second stage is recognition, or decoding, which is performed using a set of phoneme-level statistical models called hidden Markov models (HMMs). Word-level acoustic models are formed by concatenating phone-level models according to a pronunciation dictionary. These word model are then combined with a language model, which constrains the recogniser to recognise only valid word sequences. The decoder stage is computationally expensive.

Although there exist software implementations that are capable of real time performance, there are several reasons why it is worth using hardware acceleration to achieve much faster decoding. Firstly, there exist real telephony-based applications used for call-centres (e.g. the AT&T "How may I help you?" system [1]), where, the speech recogniser is required to process a large number of spoken queries in parallel. Secondly, there are non-real time applications, such as off-line transcription of dictation, where the ability of a single system to process multiple speech streams in parallel may offer a significant financial advantage. Thirdly, the additional processing power offered by an FGPA could be used for real-time implementation of the "next generation" of speech recognition algorithms, which are currently being developed in laboratories. These achieve superior performance but are much more complex and computationally expensive than current methods.

Accordingly, in this paper we describe an implementation of an HMM-based speech recognition system based on continuous HMMs, which makes use of an FPGA for the decoder stage. This work follows on from that introduced in [2].

2 Speech Recognition Theory

2.1 Hidden Markov Models and Viterbi Decoding

A hidden Markov model is a probabilistic finite state machine, which has associated with it transition probabilities - the probability of a transition from one state to another - and observation probabilities - the probability that a state emits a particular observation [3]. The probability density function can be continuous or discrete.

We define the value $\delta_t(j)$, which is the maximum probability that an HMM is in state j at time t. It is equal to the probability of the most likely partial state sequence which emits observation sequence $O = O_0, O_1 \ldots O_t$, and which ends in state j. It can be shown that this value can be computed iteratively as:

$$\delta_t(j) = \max_{0 \le i \le N-1} [\delta_{t-1}(i) a_{ij}] \cdot b_j(O_t) , \qquad (1)$$

where i is the previous state (i.e. at time t–1).

This value determines the most likely predecessor state $\psi_t(j)$, for the current state j at time t, given by:

$$\psi_t(j) = \arg\max_{0 \le i \le N-1} [\delta_{t-1}(i) a_{ij}] . \qquad (2)$$

At the end of the observation sequence, we backtrack through the most likely predecessor states in order to find the most likely state sequence. Each utterance has an HMM representing it, and so this sequence not only describes the most likely route through a particular HMM, but by concatenation provides the most likely sequence of HMMs, and hence the most likely sequence of words or sub-word units uttered.

Implementing equations (1) and (2) in hardware can be made more efficient by performing all calculations in the log domain, reducing the process to additions and comparisons only - ideal when applied to an FPGA. The resulting system structure is shown in Fig. 2.

2.2 Computation of Observation Probabilities

Continuous HMMs compute their observation probabilities based on feature vectors extracted from the speech waveform. The computation is typically based on uncorrelated multivariate Gaussian distributions [4]. These calculations can be performed in the log domain, resulting in the following equation:

Proceedings of the 10 th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'02)
1082-3409/02 $17.00 © 2002 IEEE

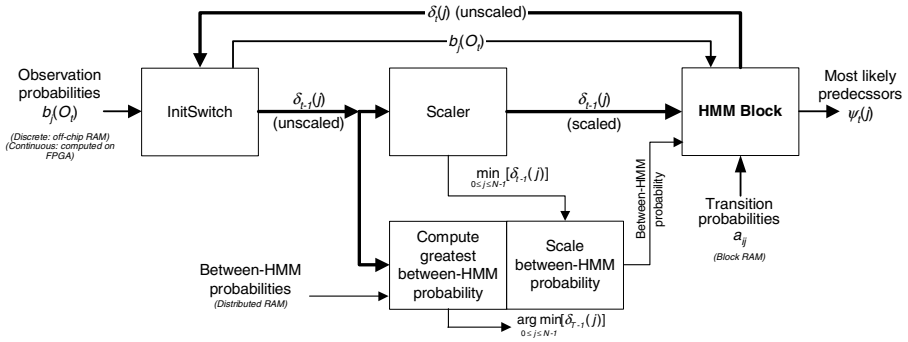

Fig. 1. Viterbi decoder core structure

$$\ln(N) = \left[-\frac{L}{2}\ln(2\pi) - \sum_{l=0}^{L-1}\ln(\sigma_{jl}) \right] - \sum_{l=0}^{L-1}(O_{tl} - \mu_{jl})^2 \cdot \left[\frac{1}{2\sigma_{jl}^2} \right] \quad (3)$$

where $\mathbf{O}_t$ is a vector of observation values at time t; μ_j and σ_j are mean and variance vectors respectively for state j; O_{tl}, μ_{jl} and σ_{jl} are the elements of the aforementioned vectors, enumerated from 0 to L–1.

3 Implementation and Results

3.1 System Hardware and Software

The design was implemented on a Xilinx Virtex XCV1000 FPGA, sitting on Celoxica's RC1000-PP development board [5]. The RC1000 is a PCI card, whose features include the FPGA, and 8 Mb of RAM accessible by both it and the host PC. The RC1000 was used within a PC with a Pentium III 450 MHz processor.

This pre-processing is performed using the HTK speech recognition toolkit [5]. HTK was also used in order to verify the outputs of our system.

The speech waveforms used for the testing and training of both implementations were taken from the TIMIT database [6], a collection of speech data designed for the development of speech recognition systems. Both the test and training groups contained 160 waveforms, consisting of 10 male and 10 female samples from each of 8 American English dialect regions.

For this implementation, we used 49 monophone models of 3 states each, with no language model.

3.2 Implementation

We implemented in software and hardware a continuous HMM-based speech recogniser, which involved computing the observation probabilities as defined in equation (3). The software was written so as to be as functionally similar as possible to the hardware implementation.

The continuous observation vectors extracted from the speech waveforms, and the mean and variance vectors for each state, consisted of 39 single-precision floating-point values.

The design occupied 5,590 of the XCV1000's slices, equal to 45%, and ran at 44 MHz.

Table 1. Results from continuous HMM implementation

	FPGA resources	Correct ness	Time/ obs (µs)	Speedup v S/W	Speedup v real time
S/W	-	56.8%	5390	-	1.86
H/W	45%	56.8%	134	40.2	74.6

3.3 Results

The results are shown in Table 1. Correctness is the number of correctly identified phones divided by the total number. Time per observation for the hardware is defined as the time between the PC releasing the shared RAM banks after writing the observation data, and the FPGA releasing the banks after writing all the predecessor information.

The hardware implementation produced identical results to the HTK software. The correctness values are clearly lower than those found in commercial speech recognition products (typically above 97%). This is because such products use significantly more complex models. Work is in progress to embed our FPGA based solution within more complex models, which should lead to a recognition rate comparable to commercial recognisers, but at much higher speed.

4 Conclusions

We have demonstrated a speech recognition system on an FPGA development board based on continuous HMMs, using a simple monophone model that is capable of performing speech recognition at a rate 75 times faster than real time.

References

[1] AL Gorin, G Riccardi and JH Wright, "How may I help you?", *Speech Communication* 23, (1997) pp 113-127.

[2] Melnikoff, S.J., Quigley, S.F. & Russell, M.J., "Implementing a hidden Markov model speech recognition system in programmable logic," *FPL 2001*, *LNCS #2147*, 2001, pp.81-90.

[3] Rabiner, L.R., "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proc. IEEE*, **77**, No.2, 1989, pp.257-286.

[4] Holmes, J. N. & Holmes WJ, "Speech synthesis and recognition," Taylor & Francis, 2001

[5] Woodland, P.C., Odell, J.J., Valtchev, V. & Young, S.J. "Large vocabulary continuous speech recognition using HTK," *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP '94)*, 1994, pp.125-128.

[6] http://www.ldc.upenn.edu/Catalog/LDC93S1.html