UNIVERSITY OF SOUTHAMPTON

Speech Recognition on Embedded Hardware

by

Ricardo da Silva

Technical Report

Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

March 31, 2013

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

by Ricardo da Silva

This report presents a proof of concept system that is aimed at investigating methods of stoftrds that are currently under development at the University of Southampton, and implements one part of a speech recognition system using an FPGA and a Linux applications processor.

Contents

Acknowledgements

Thanks to Steve Gunn, Srinandan Dasmahapatra

Chapter 1

Introduction

1.1 Goals

At the highest level, the primary goal of this project is to implement part of a modern HMM based speech recognition system, with the constraint that it must be done on embedded hardware. In pursuing this goal, the aim is to achieve several other goals that will be beneficial to the author and to the University of Southampton.

1.1.1 Speech Recognition

A complete HMM based speech recognition system is extremely complex, and can take years to design and optimise for a particular implementation. The goal of this project is not to implement one such system, but rather to explore the possibilities of what may be achieved with a low power applications processor and a relatively small FPGA. In particular, the goal is to use the applications processor (running Linux) to perform pre-processing of speech data, and use the FPGA to perform mathematical manipulations of the observations. It is hoped that the system developed here may be later used as a basis for future research into the subject.

1.1.2 The Micro Arcana

The Micro Arcana is a new hardware platform aimed at undergraduate students, currently under development by Dr Steve Gunn. In terms of hardware, one of the project goals is to further development of the Micro Arcana family of boards, and provide a valuable use case example as described in the Motivation section. Part of the project is setting up and configuring these two boards, so that they may be easily picked up by undergraduates. In addition, the aim is to build the entire system on these two boards, making it a self-contained embedded design and showing how they may be usefully combined.

1.1.3 Theoretical understanding

A major goal of the project is to develop a higher level of understanding of the algorithms used in speech recognition, and to get experience designing a large-scale embedded application. This complements the interests of the author and the subjects being studied, in particular, Intelligent Algorithms and Digital Systems Design.

1.2 Motivation

Speech recognition is an interesting computational problem, for which there is no fool-proof solution at this time. Recently the industry for embedded devices and small-scale digital systems has expanded greatly, but in general these devices do not have the power or speed to run speech recognition. Field Programmable Gate Arrays (FPGAs) may present a way of increasing the capability of such systems, as they are able to perform calculations much faster than traditional microprocessors. The author's personal interest in embedded systems, combined with the challenges of a complex system such as speech recognition, makes this an appealing area to explore.

As hardware platforms go, most of the Micro Arcana is still very new and untested, as it is still under development. In addition, there are not many examples of how they may be used, and very little documentation. In order to improve their reception by students, it would greatly help to have proven use cases and examples of how these boards may be used individually and together. Using a larger FPGA (such as an Altera DE board) was considered during the planning stages of this project, but it was decided that part of the challenge was to develop and use the Micro Arcana.

1.3 Contributions

The project implements one part of a modern speech recognition system, using two development boards from the Micro Arcana family. It is designed to be a proof of concept exercise, in order to explore the capabilities of the boards, and expand the author's knowledge of the relevant systems. Specifically, the project required substantial research into HMM based speech recognition systems, embedded Linux, and digital design. The resulting system, described in detail in Chapter 4, uses an FPGA to perform the most computationally expensive part of HMM based recognisers – scoring the states of each HMM model for a given input vector. Essentially, the ARM based "L'Imperatrice" is used as the application controller, and is connected to the FPGA based "La Papessa" board which performs the CPU intensive mathematical calculations. The embedded Linux processor reads WAV speech files, performs necessary pre-processing and sends the observation vectors to the FPGA. Given this vector, the FPGA will process it and send back scores for each state in the speech model, which

represent the probability of that state creating the observation. Given these scores, the next step for a speech recogniser would be to perform Viterbi decoding in order to find the most probable sequence of HMMs, and thus eventually find the most probable word or phoneme sequence spoken.

Chapter 2

Background

2.1 Speech Recognition Systems

In general, 'Speech Recognition' refers to the process of translating spoken words or phrases into a form that can be understood by an electronic system, which usually means using mathematical models and methods to process and then decode the sound signal. Translating a speech waveform into this form typically requires three main steps [7]. The raw waveform must be converted into an 'observation vector', which is a representative set of data that is compatible with the chosen speech model. This data is then sent through a decoder, which attempts to recognise which words or sub-word units were spoken. These are then sent through a language model, which imposes rules on what combinations of words of syntax are allowed. This project focusses on implementing pre-processing, and the first stage of the decoder, as this is an interesting task from an electronic engineering point of view.

There are a variety of different methods and models that have been used to perform speech recognition. An overview of the most popular will be described here, along with the chosen approach.

2.1.1 Tor's Algorithm

The author first became interested in speech recognition when reading about "Tor's Algorithm", which is a very simple small dictionary speech recognition system [3]. This algorithm is capable of very accurate speaker dependent speech recognition for a dictionary of about ten words. It is based on a fingerprinting model where each word in the dictionary must be trained to form an acoustic 'fingerprint'. This fingerprint is based on the time variations of the speech signal after being filtered appropriately. Then recognition is reduced to finding the Euclidean distance squared between the input vector and each of the stored fingerprints. The 'best' match is the word with the smallest distance from the input. Although this system is very simplistic, it outlines two major components of any speech recognition system –

pre-processing and decoding (recognition). More complex systems essentially just use more complex speech models and pre-processing methods.

2.1.2 Dynamic Time Warping

Speech, by nature, is not constrained to be at a certain speed – the duration of words will vary between utterance, and a speech recognition system should be able to handle this. Dynamic Time Warping (DTW) is essentially the process of expanding and contracting the time axis, so that waveforms may be compared, independent of talking speed. Combined with a dynamic programming technique for finding the optimal 'warp' amount, it became a widely used approach to solving the problem of speech duration modelling [5]. One useful property of DTW is that it may offer good performance even with little training, as it only needs one word as a template [7]. Conversly, the performance of DTW based systems cannot be increased much with more training, unlike Hidden Markov Models.

2.1.3 HMMs

By far the most prevalent and successful approach to modern speech recognition uses Hidden Markov Models (HMMs) for the statistical modelling and decoding of speech [4]. The flexibility inherent in HMMs is key to their success, as a system can be made more and more accurate by simply improving the HMM models or training the models further. The classic tutorial paper by Rabiner ([9]) is one of the best references for HMMs in speech recognition, and provides a very good overview of modern systems. The following sections are based heavily on [9] and [12].

The simplest HMM based systems use a single HMM for every word in the recognition dictionary. Given a set of observations, each HMM can be scored based on the probability that it would output the observations. The HMM with the highest score is taken as the recognised word. The most apparent limitation of this system is that a very large amount of training would be required if a dictionary of any size was to be used. At the very least, one sample of each word would need to be recorded to train the full system, which would be a very time consuming process. However, for simple applications (voice dialling, for example) this is manageable.

The next step up in complexity from single word HMMs is models that consider sub-word utterances (phonemes). This allows a smaller set of HMMs to be used for much larger dictionary recognition, as words are recognised based on sequences of sub-word HMMs. Thus instead of searching through a single HMM to recognise a word, the recognition process becomes a search through a trellis of multiple HMMs in order to find the best path through them. The most simple HMM system of this form is based on mono-phones, of which there are about 50 in English. This may be improved by using bi- or tri-phone HMMs, which model

transitions between two or three mono-phones. Using this form of HMM will increase the acoustic model size greatly however, as there are many possible combinations of mono-phones in the English language.

Even more complexity (and, potentially, recognition accuracy) can be introduced by using bi- or tri-phone HMMs, which model transitions between two or three mono-phones. Using this form of HMM will increase the acoustic model size greatly however, as there are many possible combinations of mono-phones in the English language. However, it allows context dependent scoring of phonemes, including HMMs that model word endings and starts, or silences. In the Sphinx 3 recognition engine, the internal states of these HMMs are referred to as 'senones', and the term has been adopted and used extensively in this project.

For most HMM models there are three problems:

- Training the model
- Finding the probability that an HMM produced a given observation sequence
- Finding the 'best' path through a trellis of HMMs to produce a given observation sequence

The first problem is solved by using Voxforge (2.3), and the second problem is more important for word based recognisers. This project focusses on the most computationally complex part of the second problem – scoring the senones of each HMM, for every new observation frame.

In all literature encountered, the Viterbi algorithm is the preferred method for solving problem 3. It is an iterative approach to solving the optimisation problem, and has the added bonus that not much data needs to be stored during the calculation [10]. This problem is beyond the scope of the current project, but a full explanation of the Viterbi decoding process is available from [9],[7],[11].

2.2 Speech Pre-Processing

Speech signals are complex waveforms and cannot be processed without some form of feature extraction which reduces the complexity while retaining the important features. In modern speech recognition systems the two most common methods of analysing and representing speech are: [6]

- Linear Predictive Coding (LPC)
- Mel-Frequency Cepstrum Coefficients (MFCC)

Both these methods attempt to model the movement and dynamics of the human vocal tract and auditory perception. LPC is more suited to speaker recognition (the process of identifying voices, rather than speech), while MFCCs are more useful for speech recognition [1].

The Mel-Frequency Cepstrum is based on a filterbank analysis with a cepstral transformation, which is required due to the high correlation between filterbank amplitudes. The human ear perceives sound on a non-linear frequency scale, and one way of improving recognition performance is by using a similar scale for analysis of speech. A filterbank analysis can be used to perform this discrimination between different frequencies, and the frequency bins are usually spaced using the Mel frequency scale. However, the filterbank amplitudes are highly correlated, which greatly increases the computational complexity of the HMM based recogniser as the covariance matrix will not be diagonal. In order to correct this, a discrete cosine transform is taken on the log filterbank amplitudes, finally resulting in a set of Mel Frequency Cepstrum Coefficients. The HTK (2.3) defaults to using twelve MFCC filterbank bins. [12] [7]

In order to attain MFCCs, a sampling rate must be chosen such that enough data is gathered while allowing sufficient processing time. In addition, to perform Fourier transforms on the speech, the incoming signal must be windowed appropriately. The HTK has a set of default values for these parameters, which are assumed to be appropriate.

An improvement to both LPC and MFCCs is to compute time derivatives in the feature extraction process, which gives a better idea of how the signal changes over time. In addition, the log energy of each sample may also be computed to also boost recognition ability.

2.3 The HTK and VoxForge

The Hidden Markov Model Toolkit (HTK) is a set of tools and libraries for developing and testing Hidden Markov Models (HMMs), primarily for speech processing and recognition tools [12]. Given a model structure and a set of transcribed speech recordings (a 'speech corpus'), a set of HMMs may be trained using the HTK. This includes performing all pre-processing in a number of formats, and testing recognition capabilities of a model.

Voxforge is an open source speech corpus which is aimed at facilitating speech recognition development. It provides pre-compiled acoustic models – essentially large sets of HMMs – in the format created by HTK, licensed under the GPL (GNU General Public License) [2]. The alternative would be to use another speech corpus (such as TIMIT), and then use the HTK to design and train the acoustic model. This is potentially a very time consuming process, so Voxforge is useful because it essentially cuts this step out. In addition, the Voxforge models may be easily adapted to a specific person's voice using only a few minutes of transcribed speech. The Voxforge model is very complex, with 8000 tri-phone context-dependent HMMs

with multivariate Gaussian output probabilities (25 components per distribution). Implementing a recogniser system based on this model requires a lot more work than if a simpler model was used, such as one based on discrete (output probability) HMMs. However, modern speech recognisers are likely to use a model that is as complex, if not more so.

2.4 FPGAs

2.5 Embedded hardware and Speech Silicon

A wide range of speech recognition software (commercial and open source) exists for desktop PCs or laptops. However, speech recognition for embedded systems is less widespread. Recently there has been increased research into the use of DSPs and FPGAs for speech recognition [7], [10], [8]. Of particular interest is Stephen Melnikoff's PhD Thesis, and the Speech Silicon architecture. The former investigates a variety of HMM based recognisers on an FPGA, using a PC to perform pre and post processing. The latter details a flexible FPGA based system capable of performing recognition on medium-sized vocabularies.

Both Melnikoff and Speech Silicon perform an in-depth analysis of an entire speech recognition system based on programmable logic. Both of them require relatively large FPGAs, and

2.6 Personal Contribution

The initial goal of the project, to build a complete speech recognition system, was very ambitious and had to be narrowed down as more was learnt about the complexities of modern speech recognition systems. Instead of attempting to build a full system, it was decided that development would focus on the decoding stage of recognition. Speech pre-processing is a very established process, and implementing it is usually a case of linking together the appropriate libraries.

The result of this project is a system spread across two development boards from the Micro Arcana: L'Imperatrice and La Papessa. This involved... -

- Benefit from developing my knowledge of SV, speech recognition, intelligent algorithms, etc - Providing an application for the Micro Arcana family, and showing that it can do stuff - Providing a solid basis for future work in the area

Chapter 3

Design Approach

3.1 System Overview

3.2 Analysis of Solution

Chapter 4

Design Detail

Chapter 5

System Testing

Chapter 6

Project Management

Chapter 7

Conclusions and Future Work

Bibliography

[1] Personal interview with srinandan dasmahapatra, 11 2012.

[2] Voxforge, 2012. URL http://www.voxforge.org/.

[3] Tor Aamodt. A simple speech recognition algorithm for ece341, 04 2003. URL http://www.eecg.toronto.edu/~aamodt/ece341/speech-recognition/.

[4] SJ Cox and G. Britain. *Hidden Markov models for automatic speech recognition: theory and application*. Royal Signals & Radar Establishment, 1988.

[5] Sadaoki Furui. *Digital Speech Processing, Synthesis, and Recognition*. Marcel Dekker, 1989.

[6] S.K. Gaikwad, B.W. Gawali, and P. Yannawar. A review on speech recognition technique. *International Journal of Computer Applications IJCA*, 10(3):24–28, 2010.

[7] S.J. Melnikoff. *Speech recognition in programmable logic*. PhD thesis, University of Birmingham, 2003.

[8] S. Nedevschi, R.K. Patra, and E.A. Brewer. Hardware speech recognition for user interfaces in low cost, low power devices. In *Design Automation Conference, 2005. Proceedings. 42nd*, pages 684–689. IEEE, 2005.

[9] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[10] J. Schuster, K. Gupta, R. Hoare, and A.K. Jones. Speech silicon: an fpga architecture for real-time hidden markov-model-based speech recognition. *EURASIP Journal on Embedded Systems*, 2006(1):10–10, 2006.

[11] Saeed V. Vaseghi. *Advanced Digital Signal Processing*. John Wiley and Sons, fourth edition, 2008.

[12] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Cambridge University Engineering Department, 2009.