Proceedings of the 6th WSEAS International Conference on Multimedia Systems & Signal Processing, Hangzhou, China, April 16-18, 2006 (pp204-207)

Isolated Word Recognition on an Embedded System

Mahmut MERAL, Muhittin GÖKMEN
Computer Engineering Department
İstanbul Technical University
Maslak, 34469 İstanbul, TURKİYE

Abstract: - In this paper, an embedded isolated word recognition system, which is designed for command and control applications, is presented. The hardware is designed by using a 32 bit general purpose ARM7TDMI cored microcontroller and required peripherals. Vector Quantization (VQ) and Discrete Hidden Markov Model (HMM) based speech recognition algorithms are implemented on the embedded system. The system is tested on a data set containing fifty four Turkish words taken from forty speakers. Using this data set, the VQ classifier obtained 99% and HMM classifier obtained about 97% average recognition rates in simulations.

Key-Words: - Embedded systems, Isolated word recognition, Vector quantization, Hidden Markov models

1 Introduction

Speech is the most natural way of communication between humans. Therefore automatic speech recognition has been studied for a long time to design natural human-machine interfacing systems [1, 2]. Speech has many advantages over other means of communication. People can speak roughly five times faster than they can type. During speech hands and eyes are free for other tasks. Most importantly, no practice is needed for using speech recognition interfaces so every common person can use it in machine interaction [3].

Embedded speech recognition has become a hot topic in recent years in the field of consumer electronics. Speaker dependent small vocabulary name dialing and speaker independent command controlling have been applied on hand held devices and toys [4].

Although successful speech interfaces exists for years, applying these systems on the constrained recourses of embedded systems such as memory and processing capabilities presents new problems [4, 5]. There are mainly three alternatives to implement automatic speech recognition (ASR) on an embedded system. First one is to use application specific integrated circuit (ASIC) which offers high integration, less power consumption and lowest price if produced in high quantities. Another solution is to use digital signal processors (DSP). DSP offer high processing capabilities to ensure real time constraints with the cost of high prices. Final way is to use a general propose microprocessor with required peripherals. Novel microcontrollers offer on chip peripherals such as ADC, DAC and memory to provide high integration for low prices. Furthermore microprocessor based ASR solutions provide more flexibility in terms of algorithm modification [5].

In this work, an isolated word recognition system to be used in command and control applications, is implemented on a 32 bit general purpose RISC microcontroller.

The hardware properties of the system are described in the second section. In section three, the speech recognition algorithms implemented on the embedded target is discussed. In the last chapter the result obtained from the simulation is presented.

2 Hardware

This chapter explains the details of hardware design of the embedded system. The hardware is based on a relatively low performance microcontroller, ARM7TDMI. The rest of the hardware design is focused on mixed signal processing for voice recording and prompting. The embedded system consists of two parts: digital and mixed signal systems. General overview of the system is represented in the Fig 1. The detailed discussion of the parts will be given in the subsections.

2.1 Digital System

Digital system contains a processor, memory and the other required digital peripherals. A development kit, Logic Zoom SDK, assembling Sharp LH75401 micro controller is used to fulfill digital system requirements [6, 7].

The ARM7TDMI core is a member of the ARM family of general purpose 32 bit microprocessors [8]. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles. It offers high performance for low power consumption. As a debugging assistant for developers, the EmbeddedICE-RT logic provides integrated on chip debug support for the ARM7TDMI core. It is used to program the conditions under which a breakpoint or watchpoint can occur. The ARM7TDMI processor has two instruction sets: the 32 bit ARM instruction set and the 16 bit Thumb instruction set. ARM Thumb offers high performance and high code

density by implementing 16 bit instruction set on 32 bit architecture.

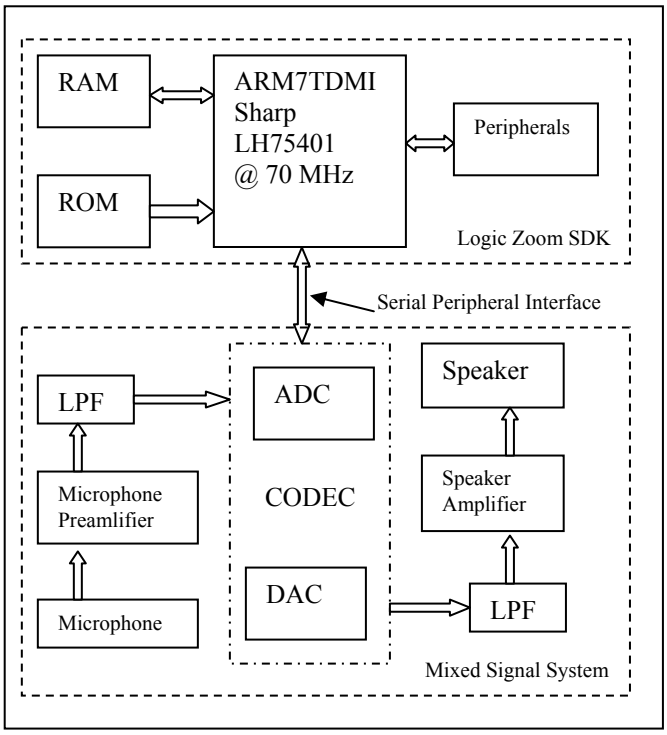

Fig 1. Schematic diagram of the hardware

2.1 Mixed Signal System

The hardware design activities are mostly focused on the mixed signal system. A board is designed for mixed signal processing and interfaced with the digital system via serial communication peripheral.

2.1.1 Microphone & Microphone Preamplifier

An electret condenser microphones (ECM) is used in the hardware. ECMs generate very weak signals, so they contain an internal preamplifier. Although EMC has its own internal preamplifier, its dynamic range is still narrow to be read by an ADC preciously. For instance EMC output signal is about 50 mV peak to peak, but ADC reads 2 V peak to peak in a 3.3 V supplied system. So that microphone output signal should be at most forty times amplified in order to efficiently use the ADCs precision range. A simple op-amp preamplifier is used in the hardware.

2.1.2 Anti-aliasing Filters

According to sampling theorem any sampled band limited signal can be reconstructed if it is sampled at rate at least twice the highest frequency contained in the signal according to Shannon's sampling theorem. But when the requirement is failed, aliasing occurs.

One way to avoid the aliasing problem is to use a low pass filter before the sampling stage for removing any frequency components above the Nyquist Rate. Anti-aliasing filters are generally implemented as analog circuits. An analog anti-aliasing filter can be a simple resistor-capacitor couple or an active circuit designed with operational amplifiers [9]. A fifth-order anti aliasing filter designed by using op-amps provided in the hardware.

2.1.3 Analog to Digital Converter

The electrical signal from the microphone is digitized prior to the speech signal processing. Digitization process is referred as analog to digital convertion and consists of three stages: sampling, quantizing and coding [10]. In speech processing applications sampling rate is varied around 6 to 16 kHz. In this work 8 kHz sampling rate is used with a 4 kHz anti-aliasing filter prior to ADC. A 16 bit linear codec is used in the hardware [11]. The codec includes delta sigma ADC, DAC and synchronous serial interface (SSI). SSI provides a simple and glue-free interface with the microcontroller.

2.1.4 Digital to Analog Converter

In our system digital to analog convertion is used for voice prompting. A low pass filter which is called reconstruction filter is necessary after the DA conversation to remove the distortion. This filter must satisfy the same conditions as in AD conversation. The output of a DAC is not strong enough to drive a loud speaker. Thus speaker amplifiers are used to amplify DAC output [12].

3 Speech Recognition Software

In this chapter, software implementation details are described. Software implementation is performed in two stages. In the first stage all the software is designed on PC. This contains the implementations of signal analysis, codebook generation for VQ, HMM parameter estimation and the classification with VQ and HMM. First all software is written with floating point arithmetic with the C++ programming language. Then the parts which are needed to be implemented on the embedded system are rewritten using fixed point arithmetic in C. This software is also used as a simulator to obtain the test results. In the second stage the necessary algorithms, that are signal analysis, VQ and HMM classifiers, are recoded according to RISC microcontroller optimization criteria [13, 14].

3.1 Speech Signal Analysis

The aim of speech analysis is to extract the speech features that we will use in pattern classification phase. Speech signal is a product of dynamic process, so that the use of short time signal processing techniques is essential. Additionally using short term techniques has another reason that processing sources can only hold and

process finite amount of data at a time [2]. Speech signal processing system consists of pre-emphasis, framing, windowing and feature extraction modules.

3.1.1 Pre-emphasis

Speech signal is pre-emphasized to spectrally flatten [1]. Generally a first order FIR filter is used for this purpose:

$$H(z) = 1 - az^{-1} \qquad 0.9 <= a <= 1 \qquad (1)$$

where a is mostly chosen as 0.9375 = 15/16 in fixed point implementations.

3.1.2 Frame Blocking and Windowing

Since speech signal is time variant, it is split into frames that it is assumed to be stationary before processing. 20 ms frames are adequate when human speech production rate is taken into account. Frames are overlapped to keep the smoothness of the spectral estimates of the adjacent frames.

Windowing is used to avoid discontinuities at the ends of the frames. Hamming window is used in windowing.

3.1.3 Linear Predictive Coding (LPC)

The LPC characterizes a stationary frame of speech by an all pole model. Assumption behind the LPC model is that the sample can be approximated with the linear combination of past p speech samples. The transfer function of the system is as follows:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \qquad (2)$$

where p is the system order and a_i's are system parameters [1].

3.1.4 LPC Cepstrum

Cepstrum is the Fourier transformation of logarithmic magnitude spectrum that is widely used in speech recognition. It is possible to obtain cepstral parameters which are called LPC cepstrum by manipulating LPC parameters. These parameters are shown to outperform LPC parameters in speech recognition systems [1].

3.2 Vector Quantization

Vector quantization can be easily applied to speech recognition in the sense of source coding [1]. Assume that there are M utterances to be recognized. Each utterance is considered as an information source. Then M codebooks are designed using the minimum average distortion objective. When a test signal is represented to classifier, it is classified as the vector quantizer that yields the minimum distortion. Generalized Lloyds algorithm, also known as k-Means algorithm, is used for codebook generation [1].

Since the VQ classifier is memoryless it is appropriate for simple vocabularies with phonetically distinct words. For utterances that can be distinguished only by temporal characteristics, such as "car" and "rock", and for complex utterances containing rich phonetic expressions, the method is not expected to provide satisfactory performance.

3.3 Hidden Markov Models

Hidden Markov Models are stochastic models widely used in speech recognition [1, 2]. In HMM approach speech is assumed to be output of a stochastic process. In isolated word recognition each utterance to be recognized is modeled with a HMM.

Discrete Hidden Markov Model (DHMM) is implemented in this work. DHMM is chosen since it is suitable to implement on a fixed point system due to its low computational complexity.

Parameter estimation is the most difficult problem of HMMs thus there is no known analytical solution to find parameters that maximize the probability for a given observation sequence. However there are some iterative techniques that locally maximizes the likelihood of obtaining an observation sequence for a given HMM. One widely used method called Baum-Welch method (or generalized expectation-maximization method) is used in this work [1].

Logarithmic version of Viterbi decoding is used in the recognition algorithm [1].

4 Results & Conclusion

The test results are obtained by simulating the embedded speech recognition system. The parameters of signal processing blocks are determined empirically and kept constant during all implementation stages. Voice samples are sampled at 8 kHz and splitted into 20 ms. %50 overlapping frames before processing. The feature vectors contain 8th order LPC Cepstrum and their derivatives, total of 16 elements.

4.1 Data Sets

A Turkish data set is created for training and evaluating the performance of the system. Voice samples of twenty male and twenty female speakers are recorded. Each utterance is voiced twice. The recorded utterances were chosen to be appropriate for a command control system that can be implemented on a washing machine for Turkish speaking users. Total of 54 utterances in the set are splitted into seven subsets. Each group contains the answer of a programming question such as program type or temperature. The biggest subset contains thirteen and the smallest contains five words.

4.2 Test Results

The system, on which the following test results are obtained, simulates an isolated speech recognizer. The data used in training and test of the systems are manually isolated. The classification is performed under the assumption that, the test input of the system is definitely a member of recognition set.

Two error rates are evaluated in test process. The first is evaluated on whole data set assuming that the test utterance is any one of the 54 classes. In the second, each utterance is considered in its subset and the average error rate is evaluated on seven subsets. It is because only a limited number of utterances are needed to be recognized at a time when a dialogue guided command recognition system is designed.

In table 2, the error rates obtained from VQ classifiers with various codebook sizes are presented. In Table 3, the error rates obtained from HMM classifiers with respect to number of HMM states and codebook size are given.

Table 2: Error rates obtained from VQ classifier

Codebook Size	Error Rate on Whole Set	Average Error Rate on Subsets
16	9,07%	2,31%
32	5,83%	1,30%
64	5,37%	0,93%

Table 3: Error rates obtained from HMM classifier

Number of States	Codebook Size	Error Rate on Whole Set	Average Error Rate on Subsets
4	16	25,19%	6,67%
	32	14,91%	4,63%
	64	12,41%	4,63%
5	16	25,65%	5,28%
	32	12,31%	4,44%
	64	12,22%	3,06%
6	16	20,37%	5,93%
	32	12,22%	3,70%
	64	11,57%	2,96%
7	16	23,43%	6,57%
	32	13,33%	3,70%
	64	11,39%	2,69%
8	16	20,83%	6,02%
	32	12,22%	3,98%
	64	12,31%	2,78%

The test results obtained from the simulation system show that the isolated speech recognition system is implemented successfully. The VQ classifier gives at most 99.07% and HMM classifier gives about 97.31% average recognition rates.

It is obvious from these results that VQ classifier outperforms the discrete HMM classifier. However, HMM classifier needs much less system resources than VQ classifier. A VQ classifier uses a codebook for each utterance to be recognized. So that, when vocabulary gets larger, the need for system resources increases dramatically in terms of computational and storage capacity. A HMM classifier uses only one codebook and HMM parameters require much less storage than codebooks. Moreover, Viterbi decoding algorithm in HMMs is computationally much simpler than VQ codebook search. In conclusion the choice of classifier depends on the vocabulary size and tradeoff between system resource costs and performance needs.

References:

[1] Rabiner L., Juang B., *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[2] Deller J.R., Hansen J.H.R., Proakis J.G, *Discrete-Time Processing of Speech Signals*, IEEE Press., 2000.

[3] Arnone L., Bocchio S., Rosti A., *On embedded system architectures for speech recognition applications: the gap between the status and the demand*, ISSPIT 2004, December 18-21, 562 – 566.

[4] Alewine, N., Ruback, H., Deligne, S., *Pervasive speech recognition*, IEEE Pervasive Computing, Volume 3, Issue 4, Oct-Dec 2004, 78 – 81.

[5] Devisch D., *Building Speech Recognition into Portable Products*, Electronics Engineer, December 1999.

[6] Wick J., *Zoom Starter Development Kit Quick Start Guide*, Logic Product Development, 2005.

[7] LH754xx, LH75400/01/10/11 *System on Chip Data Sheet*, Sharp Microelectronics, 2005.

[8] DDI 0210C, *Technical Reference Manual Revision: r4p1*, ARM Limited, 2004.

[9] AN928, *Filter Basics: Anti-Aliasing*, Dallas Semiconductors, 2002.

[10] Furui S., *Digital Speech Processing, Synthesis and recognition*, Marcel Darkel Inc., 2001.

[11] *PCM3500, Low Voltage, Low Power, 16-Bit, Mono VOICE/MODEM CODEC Datasheet*, Burr – Brown, 1999.

[12] *TPA301, 350-mW Mono Audio Power Amplifier Datasheet*, Texas Instruments, 2004

[13] DAI 0033A, Fixed *Point Arithmetic on the ARM - App Note 33*, ARM Limited, 1996.

[14] DIA 0034A, *Writing Efficient C for ARM - App Note 34*, ARM Limited, 1998