REAL TIME VOICE PROCESSING (VOICE/SPEAKER RECOGNITION)

D. Farrokhi, R. Togneri, Y. Zhang,
and
Y. Attikiouzel

Department of Electrical and Electronic Engineering
University of Western Australia
CIIPS Laboratory

ABSTRACT- Communication systems are one of the biggest industry in the world. Today, man's communication in the globe is greatly dependant on its communication with computers. One of the most exciting field of research which can revolutionise human interaction with computers is Voice Recognition. So far research had a reasonable success in providing algorithms for the encoding of speech data segments and the classification of these sequences of segments.

Some of the most important challenges, are those which attempt to produce real time automatic voice recognition.

Real Time Voice Recognition (RTVR) was not possible for a long time, however since the Digital Signal Processing (DSP) chips such as TMS320C30/40, i860XP and i860XR have been produced it is possible today to process the speech signal as fast as a word is spoken. This is done by embedded processing of speech signal into the daughter board for pre-processing, and doing a post and final processing in the mother-board.

In this paper, I will discuss some problems which could in any real time embedded program. One of the question, in this paper is the minimum processing speed required for pre-processing of speech data. The other question is on the post-processing module requirements. The last topic is about the over all speed requirement of the complete Real Time Voice Recognition System.

INTRODUCTION

Since the creation of the digital computer, man desired to interact with computer through speech. This was not possible for a long time, until recently when the technology advances in many aspects such as DSP chips, Mathematical Science and fast CPU's.

One of the tool which makes the RTSR a reality is cleverly designed algorithm called Hidden Markov Chain which is also known as Hidden Markov Model (HMM).

One of the algorithm which plays a key roll in a reliable voice recognition system is the VQ algorithm. The Gaussian Mixture Modelling using Expectation Maximisation (GMM/EM) algorithm is the most efficient vector quantisation algorithm, which has been tested in our research laboratory. The other vector quantisation algorithm which was tested is the well known clustering and splitting (LBG) Vector Quantisation (VQ). The other algorithm which was also tested in SUN platform and not in DOS system is the Self-Organising neural Map (SOM) using Kohonen algorithm.

Our comparison on the existing floating point DSP board shown that there are DSP board available which process signal faster than TMS320C30. However we have developed our software in TMS320C30 chip provided by Texas Instrument on a package called Evaluation Module. The reason being its good software support, and price.

REAL TIME VOICE RECOGNITION

Introduction

The real time voice recognition must be based on DOS system for obvious reasons. The system consists some pre-processing algorithms and some post processing routines.

The pre-processing consist of low pass and high pass filtering (LPF/HPF) again for the obvious reasons. The other pre-processing algorithms include, Windowing of Speech data, Real Forward Fast Fourier Transformation (R_FFFT) and first level quantisation using logarithmic formula (known as NRG or Filter Bank algorithm).

The post-processing system consist of two broad stages. First is production of quantised vectors using the well known LBG technique or the GMM/EM algorithm. Second, recognition of quantised speech data applying HMM.

Pre-processing of speech data

The useful bandwidth frequency for speech data covers between 70-4.0k Hz as in telephone line.

Although this could be argued from speech recognition point of view, but most of the new telephone signalling protocol such as R2 digital and Integrated Signalling Digital Network (ISDN) include only the audio frequencies between 150-3.3k Hz for speech data transmission.

Keeping this in mind the C30 board has been configured for 14 bits analogue to digital conversion plus setting the configurable low pass and high pass filtering chips to the cut off frequencies of 150 Hertz, and 3.6k Hertz respectively.

To minimise the effect of poor convergence, particularly in the vicinity of dis-

continuities at the start and the end of each frame the speech data must be multiplied by a window function. The window function which produced the best result in this process was Welch window function. The formula for the Welch window is the following:

Welch Window Function
1 - [j-0.5(N-1)/0.5(n-1)] * [j-0.5(N-1)/0.5(n-1)]

Other window functions which were tried in conjunction with R_FFFT are as follow:

Parzen Window : 1 - | (j - 0.5 (N-1)) /0.5(N+1) |
Hanning Window : 0.5[1-cos(2*PI*j/(N-1))]
Hamming Window : 0.54 - 0.46*cos(2.0*PI*j/(N-1))
Blackman Window : 0.42 - 0.5*cos(2.0*PI*j/(N-1)) + 0.08*cos(4.0*PI*j/(N-1))

It should be noted that N is the number of points or samples in each frame and j is the sample points from (1,2,3....N).

Research shown that, the amplitude of speech signal varies significantly with time. For example the amplitude of unvoiced segments is generally much lower than the amplitude of voiced segments. Research also indicated that speech signal like many other analogue signals is not periodic. These properties of speech signal lead the scientist to employ a Short-Time energy representation of speech signal as follows:

$$E_n = \sum_{m=-\infty}^{\infty} x^2(m) w^2(n-m)$$

where $w^2(n-m) = h(n)$ describes the characteristic of a window.

Before R_FFFT function the Welch window function was applied to each frame and the results recorded as follow:
Table 1.0 - Benchmark Real Forward FFT (EVM Board)

Number of Points	FFT Timing in ms	Comments
64	0.042	Code+data+twiddle in internal RAM
128	0.089	Code+data+twiddle in internal RAM
256	0.198	Code+data+twiddle in internal RAM
512	0.443	Code+data+twiddle in internal RAM
1024	0.994	Code+data+twiddle in internal RAM

Bit-reversal operation is not included in this benchmark.

The R_FFFT for RTVR was done on 512 points with 256 point overlaps. Given the 8k Hz sampling A/D speed the time it takes to fill a buffer of 256 points is 32 ms. Given above benchmark for 512 R_FFFT the FFT is 72 faster than the A/D speed.

This leaves comfortable time for our next processing function NRG routine. It should be noted here that windowing was one way to overcome the discontinuity and possible overflow of the program, the other was the great care in writing of the program.

Real Time Filter Bank Conversion (NRG)

There has been much discussion about the dimensionality of the speech signal. Another word, how many dimensions adequately can define the speech signal in the space. The dimension which we chosen for the real time system posses twelve dimensions. Basically Filter bank data in this system is a binning mechanism which reduces 256 magnitude FFT points to a vector of 12 dimension. Another word, it convert n-point FFFT to m-point filter bank (NRG) data. During the FFFT to NRG it also does an overlapping between each frame, as well as logarithmic mel-scaling. The overlapping is for smoothness and continuity of the operation. The mel-scaling is adapted from natural log characteristic instead of straight line behaviour.

Vector Quantisation

Introduction

VQ is a technique for source coding. It has been used to reduce the amount of computations and save on storage media, without losing important information.

LBG algorithm

One of the most common VQ algorithms is the LBG algorithm which employs K-means clustering and has the benefit of being simple with a moderate computation. Since its operation utilises one clustering step and one splitting step for each iteration, it is also called the clustering and splitting algorithm. The process starts by finding the centroid vector of input sample data, then splitting it.

A lot of literature has been produced on this subject, however the most original literature is Linde et al. in 1980 [1].

GMM/EM Algorithm

The EM algorithm is used by statisticians in a situation where some data values are missing in order to improve the quality of estimation. The data could be missing because there is problem with data collection etc. The EM algorithm in this case is applied to compute the parameters of a Gaussian Mixture Model (GMM). This was first described by Wolfe J. H. in 1970 [2]. Then Dempster et al. (1977) [3] described the GMM using Expectation Maximisation algorithm. It was shown that the EM algorithm is convergent and that the convergent rate is quadratic.

SOM algorithm

An alternative approach to VQ, based on neural network (NN) system, was proposed by Kohonen (1990) [4]. The SOM method is a closest neighbour classification scheme that uses un-supervised learning algorithm to determine the position of the codebook vectors. The SOM does not use the density function instead, it defines the class boarder according to the nearest neighbour rule.

RTVR system

Currently, the most popular approach to speech recognition is the combination of VQ for the encoding of segments of speech with HMM for the classification of sequences of segments. Therefore when the speech data is pre-processed, any of the LBG/HMM, GMM/EM/HMM, or SOM/HMM could be applied to finish the process. The performance of all the above post-processing system was discussed in an early papers, Zhang et al., 1991, 1992 [5] and (Togneri et al., 1992) [6]. In these papers, it was described how the GMM provides a better description of the speech space than either the LBG or SOM. The RTVR system was built using codebook and HMM recognisor model list created on a SUN platform using CDIGIT or NIST speech data bank. CDIGIT is a speech data bank of approximately 1000 utterances of females and males recorded in a normal environment, while the NIST data base is recorded in a noise free environment.

Porting LBG, GMM/EM vector quantisor and HMM recognisor programs into DOS system which poses a 486DX/50 CPU architecture was not effected much by speed which SUN system had. However, it is quite challenging to rewrite the program to work with a limited memory available in DOS.

The speed of these could be greatly effected if a third party program is employed to overcome the memory limitation (Huge Virtual Array/Matrix (HVA) provided by Boston Technology). Since all of the training programs mention above require around 2M Bytes of RAM it is necessary either employ an Extended Memory Manager (EMM) program or rewrite the program using HVA to overcome this problem. Although, the process is very slow with HVA, but in training codebook or creating HMM models speed is not an issue.

Recently Borland C, as well as Microsoft C providing some sort of library tools which makes it easier to use the extended memory. Speed is an issue when it comes to recognition program modules. All of the GMM/EM, LBG and HMM are not only memory hungry but also very computationally intensive. Comparing LBG with GMM/EM, LBG run faster than GMM/EM however the efficiency of the GMM/EM is higher than the LBG program.

The GMM/EM/HMM or LBG/HMM program must be completed in less than 500 milli second for a reasonable RTVR. At the present the LBG/HMM program in DOS taking around 700 ms which means a recognition within 1.7 second. Although, this could be improved by a factor of two (0.85 second) but there is limit on the speed of the operation. The structure of the existing program does not

allow any faster response than 1.7 second. This is because the CPU has to wait for either a set time until all of the sample to arrive (typically 12 * 30 sample which is around 1 second) or detect the start and the end of an utterance.

The other important consideration is the type of speech data which the LBG or GMM/EM is trained on. A rule of thumb is to train these system with a data which had been recorded in a normal environment and not in a background noise free environment.

Conclusion

We have argued that because of the critical time consideration (depending upon the length of utterances), THE GMM/EM/HMM or LBG/HMM must be faster. The other issue in RTVR in DOS was the memory limitations, this could be solved by going to alternative operating system such as OS2 or window NT. Another issue which could improve not only the speed limitation of the system but also the memory problem somewhat, is writing some of the time consuming post-processing routine in assembly. The last but by no mean the least is the issue of detection of the start and the end of an utterance.

REFERENCES

[1] Linde, Y., Bouzo, A., and Gray, R. M.(1980), "An Algorithm for Vector Quantizer Design", IEEE Trans. on Comm., COM-28(1), PP 84-95.

[2] Wolf, J. H.(1970),"Pattern Clustering by Multivariate Mixture Analysis", Multivariate Behavioural Research, 5, 329-350

[3] Dempster, A. P., Laird, N. M., and Rubin, D. B.(1977), "Maximum Likelihood from Incomplete Data via the EM algorithm", Proc. R. Stat. Soc. B,39(1), PP.1-38.

[4] Kohonen, T. (1990) 'The self-organising map', Proceeding of the IEEE, 78(9), 1464-1480.

[5] Zhang, Y. and deSilva, C. J. S.(1991) "An Isolated Word Recognisor Using EM Algorithm for Vector Quantisation", IREECON 1991, (Sydney, Australia), 289-292.

[6] Togneri, R., Farrokhi D., Zhang Y. and Attikiouzel Y. (1992) `A Comparison of the LBG, LVQ, MLP, SOM and GMM Algorithms for vector quantisation and clustering analysis`, Proceeding of The Fourth Australian International conference on Speech Science and Technology, Brisbane, Australia.