RESEARCH PAPER | Engineering | Volume : 5 | Issue : 7 | July 2015 | ISSN - 2249-555X

Optimal Approach for Real Time Continuous Speech Recognition System(Hmm)

KEYWORDS: Hidden Markov Model, Detail language search model, Gaussian Mixture Model, Simplified search model

Dr.C.Rajeshkumar

Professor Jeppiaar Institute of Technology,Chennai.

ABSTRACT *We have developed the 30K word real time continuous speech recognition based on a context dependent Hidden Markov Model (HMM). Here we are using a 30K word language model instead of previously using 20K[15] word speech recognition. It has opened new opportunities for speech recognition innovations. In 20K [15]word speech recognition has been designed with limited vocabulary .i.e., 800 words[9] but in this 30K word language model to be designed by using the high level vocabulary. In this system contains two parts. One is training and second is testing. First different input speech signals will be stored in training kit. Second will give different speech signals for testing, after comparing with training kit it will display final output. Gaussian Mixture Models (GMMs)[3] are used to represent the state of output probability of HMMs.*

INTRODUCTION

Speech recognition is defined as the ability to identify a spoken word or a sequence of words. The main idea behind the system is to first train it with several versions of the same word, thus yielding a "reference fingerprint".It is an advantage of high processing speed and low power consumption.Speech recognition based on a Hidden Markov Model (HMM) can provide high recognition accuracy, thus has been used in various applications such as automatic transcription, audio indexing, navigation, mobile devices, ubiquitous systems, and robotics. Large vocabulary real-time continuous speech recognition (LVRCSR)[8] with acoustic and language models is too resource-hungry and power-sensitive for software applications.

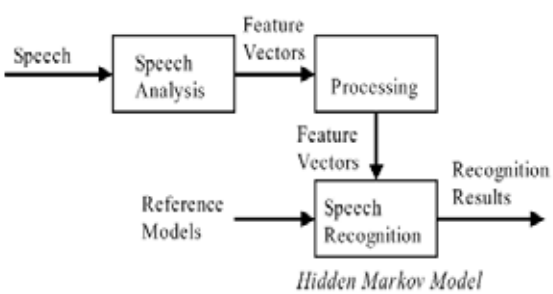

Fig 1. Basic flow chart for speech recognition

Hardware implementation by VLSI or FPGA is demanded especially for use in mobile equipment and intelligent robots because of advantageous high processing speed and low power consumption. To increase the data-pin to improve the data-transmission ability of IO, but their architecture is not extendable for a larger vocabulary because it consumes too much power and is not cost-effective: it requires multiple FPGAs. Measurement results show that our test chip can achieve 30-kWord continuous real time speech recognition with power consumption and only slight accuracy degradation.

We proposed a VLSI implementation for 30-k Word real-time continuous speech recognition. It employs algorithm optimization such as two-stage language model search to reduce cross-word transitions for the Viterbi search, beam pruning using a dynamic threshold to avoid sort processing. A variable-frame look-ahead scheme is used to reduce the memory bandwidth for GMM computation[3]. We introduced part of the External DRAM data into the internal cache memory using the locality of speech recognition and proposed specialized cache architecture to improve the cache hit rate. Elastic pipeline operation between the Viterbi search and GMM processing is applied. We analyzed the trade-off between the accuracy and the important parameters in viterbi computation to choose the most appropriate parameter combination.

2 SPEECH RECOGNITION

In speech recognition (SR)[9] is the translation of spoken words into text. It is also known as Automatic Speech Recognition (ASR). Some SR systems use training where an individual speaker reads sections of text into the SR system. These systems analyze the person's specific voice and use it to fine tune the recognition of that person's speech, resulting in more accurate transcription. Systems that do not use training are called Speaker Independent systems. Systems that use training are called Speaker Dependent Systems. Speech recognition applications include voice user interfaces such as voice dialing, call routing, domestic appliance control, search, simple data entry, preparation of structured documents, and speech-to-text processing.

2.1 OVERVIEW OF SPEECH RECOGNITION

The performance of speech recognition systems is usually evaluated in terms of accuracy and speed. Accuracy is usually rated with Word Error Rate (WER), whereas speed is measured with the real time factor. Other measures of accuracy include Single Word Error Rate (SWER) and Command Success Rate (CSR)[6]. However, speech recognition is a very complex problem. Vocalizations vary in terms of accent, pronunciation, articulation, roughness, nasality, pitch, volume, and speed. Speech is distorted by a background noise and echoes, electrical characteristics. Accuracy of speech recognition varies with the following:

- Vocabulary size and confusability
- Speaker dependence vs. independence

Volume : 5 | Issue : 7 | July 2015 | ISSN - 2249-555X

- Isolated, discontinuous, or continuous speech
- Task and language constraints
- Read vs. spontaneous speech
- Adverse conditions

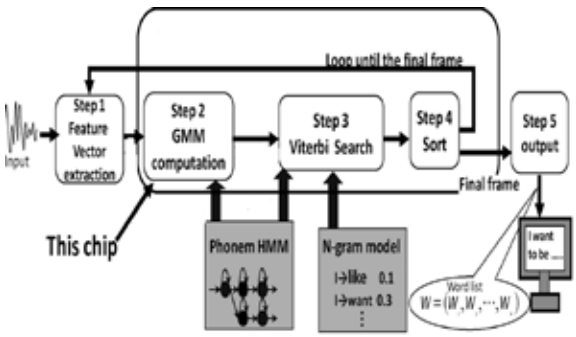

Figure 2.1 Speech recognition with HMM algorithm

Fig. 2.1 presents the speech recognition flow with the HMM algorithm. The following items describe concrete stages.

Step 1: Feature vector extraction
The input speech signal is converted from the time domain to frequency domain to obtain more unique acoustic characteristics. Feature vectors are extracted from 30 ms length of speech every 10 ms.

Step 2: GMM computation
A phonemic-model GMM is read and state output probabilities is calculated for all active state nodes.

Step 3: Viterbi search
It is calculated for all active state nodes using state output probabilities, transition probabilities, and the -gram language model.

Step 4: Sort
According to the beam width, active state nodes having a higher score are selected. The others are dumped.

Step 5: Output sentence
The word list with the maximum score is output as a speech recognition result after final-frame calculation and determination of the transition sequence.

2.2 SPEECH RECOGNITION PRINCIPLE
Speech recognition is performed by identifying a sound based on its frequency content. In order to achieve this, the frequency content of several samples of the same sound must be averaged in a training phase. Then, the frequency content of a sound input can be compared to the fingerprint by treating them as vectors and computing[4] the distance between them. If a sound is close enough to the reference, then it is considered to be a match.

2.3 SPEECH RECOGNITION FOR HMM
A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. The mathematics behind the HMM was developed by L. E. Baum and coworkers. It is closely related to an earlier work on optimal nonlinear filtering problem by Ruslan L.Stratonovich, who was the first to describe the forward-backward procedure.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a Hidden Markov Model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden'. Hidden Markov Models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part of speech tagging, musical score following, partial discharges and bioinformatics [1].

3 BLOCK DIAGRAM EXPLANATION
3.1 DESCRIPTION
Fig. 3.1, in the traditional language model search, only our second search treated every frame. However, in our proposed language model search, the second stage is treated at every frame. By applying this proposed search, the computational amount and memory bandwidth can be reduced.

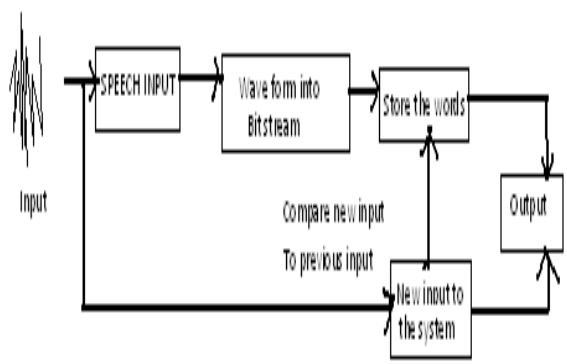

Figure 3.1 Block diagram of continuous speech recognition

In this scheme will increase continuous speech length and will reduce total power consumption of the system. Two-stage language model search scheme reduce the computational workload and memory bandwidth for cross-word transitions to isolated trees. This scheme is derived from the transition frequency difference between phonemic HMM and language HMM.

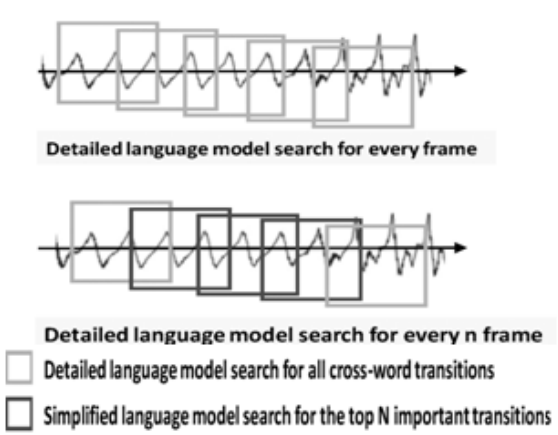

Figure 3.2 Two stage language search model

Volume : 5 | Issue : 7 | July 2015 | ISSN - 2249-555X

The cross-word transition search is divided into two stages.

- The first stage is a simplified language model search
- The second stage is a detailed language model search

3.2 SIMPLIFIED LANGUAGE SEARCH MODEL

The first stage is a simplified language model search for the top important transitions of two-gram probability. It consists of single frame to the transition of cross word as shown in figure 3.2. Here we are going to measure the peak or high values for the speech signal given as an input to the system

3.3 DETAIL LANGUAGE SEARCH MODEL

The second stage is a detailed language model search for all crossword transitions. With the increase of the detailed language model search Cycle, we can achieve greater reduction of cross-word transitions, which is the main processing undertaken in Viterbi computation. However, the risk of losing the cross-word transition to the correct candidate word might increase, thereby affecting the recognition accuracy. Moreover, the beam width and the number of cross-word transitions during the detailed search and the simplified search strongly influence the recognition accuracy. Therefore, the trade-off of these parameters described above must be discussed carefully.

First, the trade-off of the detailed language model search cycle and the number of cross-word transitions during the simplified language model search are discussed. The beam width is set to 4000. The cross-word transitions during the detailed language model search[9] are set to 2000 to maintain high recognition accuracy.Fig.3.3 presents the relation between the detailed search cycle and the number of cross-word transitions.

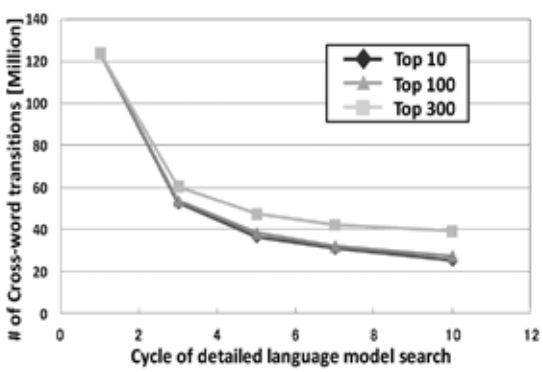

Figure 3.3 Cycle of detailed language model search versus the number of crossword transitions.

We measured the accuracy using a referential software prototype profiling with Julius 4.0. The test speech data consists of 48 test patterns, which totally include 172 sentences of Japanese speech spoken by different speakers. The average values of all the patterns for each parameter set are shown in the following graphs.

4 HARDWARE AND SOFTWARE IMPLEMENTATION

4.1 HARDWARE IMPLEMENTATION

A common way to implement a hardware viterbi decoder.A hardware Viterbi decoder for basic (not punctured) code usually consists of the following major blocks:

- **Branch metric unit (BMU)**

A branch metric unit's function is to calculate branch metrics, which are normed distances between every possible symbol in the code alphabet, and the received symbol.

- **Path metric unit (PMU)**

A path metric unit summarizes branch metrics to get metrics for 2^{K-1} paths, where K is the constraint length of the code, one of which can eventually be chosen as optimal. Every clock it makes 2^{K-1} decisions, throwing off wittingly non optimal paths. The results of these decisions are written to the memory of a traceback unit.

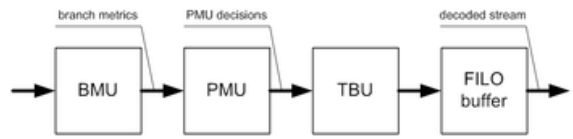

Figure 4.2 Hardware implementation of viterbi decoder

- **Traceback unit (TBU)**

Back-trace unit restores an (almost) maximum-likelihood path from the decisions made by PMU. Since it does it in inverse direction, a viterbi decoder comprises a FILO (first-in-last-out) buffer to reconstruct a correct order.

4.2.2 SOFTWARE IMPLEMENTATION

One of the most time-consuming operations is an ACS butterfly, which is usually implemented using an assembly language and appropriate instruction set extensions to speed up the decoding time.

5. RESULTS

We have to take the sample input and calculate the quantization, FFT and word fingerprint.

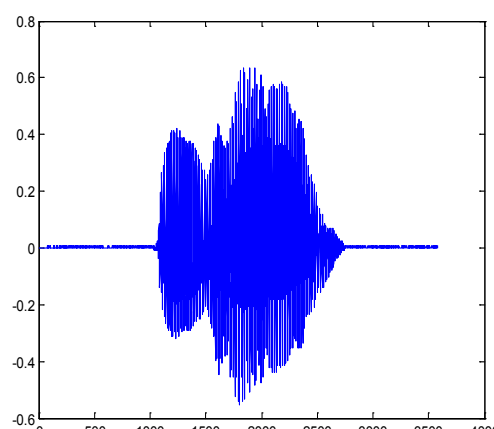

Fig5.1: Quantization of sample speech signal

Volume : 5 | Issue : 7 | July 2015 | ISSN - 2249-555X

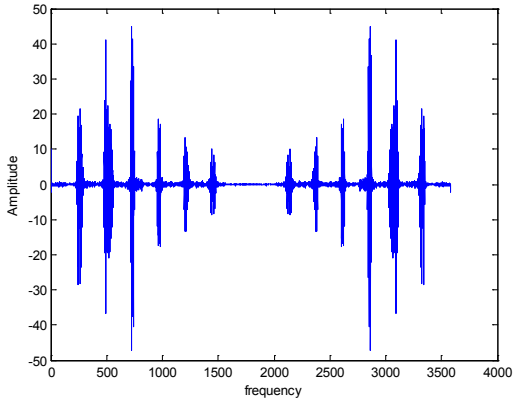

Fig 5.2 FFT of sample speech signal

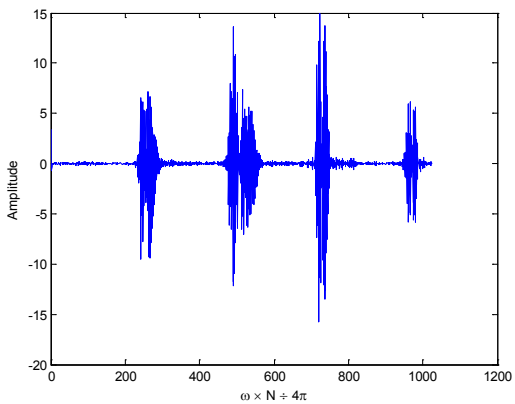

Fig 5.3 Word fingerprint for sample speech signal

Fingerprint values are stored and the new inputs are compared to precisely stored input for displaying the output. These outputs are taken in the form of speech using FPGA.

6. CONCLUSION

To achieve gains in hardware speech recognition, we must first realize the requirements and limitations of a hardware-based recognizer by prototyping the design. We address these issues in this paper and describe in detail the design and implementation of a fully functional speech recognizer The design recognizes a 1000 word vocabulary, is speaker independent, and recognizes continuous (connected) live-mode speech. Our current design runs at 50MHz, decodes at roughly 2.3 times slower real-time, achieves the same accuracy as state of- the-art software, and is, to the best of our knowledge, the most complex recognizer architecture ever fully committed to a hardware-only form. Our current work focuses on much larger vocabularies (5000 – 2O,000 words), at rates much fast than real-time, leveraging the hardware resources of a more sophisticated FPGA-based platform.

REFERENCE

1. "Adaptive beam pruning techniques for continuous speech recongnition",Hugo van hamme and flip van Aelton, Koning Albert1 laan,64 B1780 wemmel Belgium. | 2. "A Gaussian Mixture Model Spectral Representation for Speech Recognition", Matthew Nicholas StuttleHughes Hall and Cambridge University Engineering Department, Jully 2003. | 3. B. Pellom, R. Sarikaya, and J. Hansen, "Fast likelihood computation techniques in nearest-neighbor based search for continuous speech recognition," IEEE Signal Process. Lett. vol. 8, no. 8, pp. 221–224, Aug. 2001 | 4. E. C. Lin, Y. Kai, R. A. Rutenbar, and T. Chen, "A 1000-word vocabulary, speaker-independent, continuous live-mode speech recognizer implemented in a single FPGA," in Proc. 15th ACM/SIGDA Int. Symp. FPGA, Monterey, CA, Sep. 2007, pp. 60–68. | 5. H. Ney and S. Ortmanns, "Dynamic programming search for continuous speech recognition," IEEE Signal Process. Mag., vol. 16, no. 5, pp. 64–83, Sep. 1999. | 6. K. Yu and R. A. Rutenbar, "Profiling large-vocabulary continuous speech recognition on embedded devices: A hardware resource sensitivity Analysis," in Proc. 10th Conf. Interspeech, Brighton, U.K., Sep. 2009, pp. 995–998. | 7. K. You, Y. Choi, J. Choi, and W. Sung, "Memory access optimized VLSI for 5000-word speech recognition," J. Signal Process. Syst., vol. 63, no. 1, pp. 95–105, Apr. 2011. | 8. "Lexicon adaptation for LVCSR: speaker idiosyncrasies, non-native speakers, and pronunciation choice", Wayne Ward, Holly Krech, Xiuyang Yu, Keith Herold, George Figgs, Ayako Ikeno, Dan Jurafsky, Center for Spoken Language Research University of Colorado, Boulder. | 9. "Speaker verification using adapted Gaussian mixture models", Douglas A. Reynolds ,Thomas F.Quatieri and Robert S.Dunn ,MIT Lincoln Laboratory, 244 Wood St., Lexington, Massachusetts 02420,Digital Signal Processing 10, 19–41 (2000), http://www.idealibrary.com on. | 10. "Subspace Gaussian Mixture Models for Large Vocabulary", Speech Recognition, Liang Lu, the university of Edinburgh, May 14, 2010. | 11. "Viterbi decoder with reduced metric computation", the application claims priority of provisional application No.60/009337 field, Dec.29, 1995. | 12. S. Yoshizawa, N. Wada, N. Hayasaka, and Y. Miyanaga, "Scalable architecture for word HMM-based speech recognition and implementation in complete system," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 1, pp. 70–77, Jan. 2006. | 13. Y. Choi, K. You, and W. Sung, "FPGA-based implementation of a real-time 5000-word continuous speech recognizer," in Proc. 16th Eur. Signal Process. Conf., Lausanne, Switzerland, Aug. 2008. | 14. Y. Choi, K. You, J. Choi, and W. Sung, "VLSI for 5000-word continuous speech recognition," in Proc. IEEE ICASSP, Taipei, Taiwan, Apr.2009, pp. 557–560. | 15. Y. Choi, K. You, J. Choi, and W. Sung, "A real-time FPGA-based 20,000-word speech recognizer with optimized DRAM access," IEEE Trans.Circuits Syst. I, Reg. Papers, vol. 57, no. 8, pp. 2119–2131, Aug. 2010. |