EMBEDDED SPEECH RECOGNITION SYSTEM ON 8-BIT MCU CORE

Dong Wang, Liang Zhang, Jia Liu and Runsheng Liu

Department of Electronic Engineering, Tsinghua University, Beijing
wdong01@mails.tsinghua.edu.cn

ABSTRACT

This is a small vocabulary, speaker independent, discrete word speech recognition system based on the System On Chip (SOC) philosophy. It is implemented on an 8-bit MCU (Micro Control Unit). The system adopts Linear Predictive Cepstral Coefficient (LPCC) related features followed by Vector Quantization (VQ) step as front end, and Hidden Marcov Model (HMM) as speech model. Confidence measure based on likelihood score (LLS) are given for rejection of Out-Of-Vacabulary (OOV) words. The recognition rate is improved with corrective training, and robustness is acquired by integrating confidence measure into the system. The recognition accuracy is nearly 97% with a vocabulary up to 30 phrases under normal conditions. Simple speech codec is also implemented for all speech I/O purpose.

1. INTRODUCTION

The embedded speech recognition system is becoming more important in the recent years with the rapid development of the handheld devices and other portable devices. Handheld devices are extremely limited by cost, power and size. And it needs more smart user interface other than keyboard. In the field of consumer electronics such as toy industry which only requires simple speech commands, small-vocabulary speaker-dependent name dialing and speaker-independent command control are very useful. And implementions based on Dynamic Time Warping (DTW) and Continuous HMM (CHMM) have been successfully employed in mobile phones, personal digital assistants and toys. But currently most of these implementations are on DSP platform, which is very expensive, especially for cost sensitive toys. For example, QualComm has developed PureVoice speech recognition engine for a mixed 100 word speaker-dependent and 30 word speaker independent task in one system within the CDMA chipset. [1] Using Artificial Neural Network (ANN) schema, Sensory Inc. achieves an recognition rate of above 95% for speaker independent tasks with up to 15 words using 8-bit MCU. [2] Formally we also proposed a speaker dependent system based on an 8051 core in [3].

In this paper, we use a different Discrete HMM (DHMM) approach for the small vocabulary, speaker independent, discrete word task on 8-bit MCU core. The subtle balance of cost and performance is achieved by choosing the 8-bit microcontroller with 16-bit co-processor platform, and by carefully adjust the training and recognizing algorithms we use, and by using some heuristic methods. We choose DHMM because it requires very few hardware resources. The hardware architecture makes flexible vocabulary and low power consumption possible. To achieve a user friendly all speech interface, speech codec of Continuous Variable Slope Delta modulation (CVSD) is used. The bit rate is 16kbits/s.

The database we used is recorded under normal room conditions. 90 isolated Mandarin words are spoken 5 times by 100 persons, among whom 50 are male and 50 female. The speech utterances of 40 men and 40 women are randomly selected for training, the left for testing.

The rest of this paper is organized as follows. In section 2, we describe the hardware architecture of the speech recognition SOC. In section 3, the software architecture and the algorithmic details are introduced and relevant results are presented. The overall evaluation is given in section 4 and the system performance is summarized in section 5.

2. HARDWARE ARCHITECTURE

The chip we choose is S3CB519 [4], which is composed of an 8-bit MCU core with 3K bytes on-chip RAM and 16K words on-chip ROM, 16-bit co-processor, a codec of 14bit ADC and 8 bit DAC, PWM (Pulse Wide Modulation), 6 general purpose I/O ports and other peripheral circuits. The maximum system clock frequency is 8.2MHz. The block diagram is shown in Figure 1 and details of each block are described below.

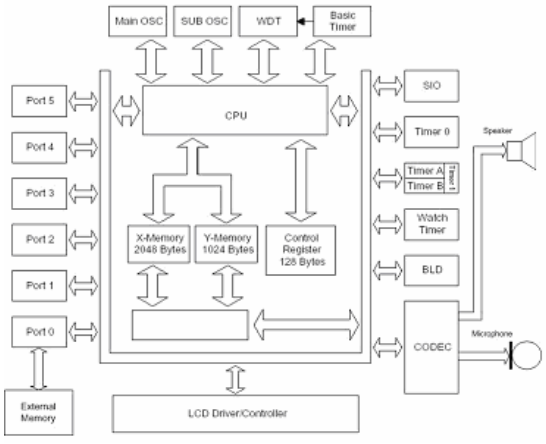

Fig. 1 Hardware block diagram

The MCU core is an 8-bit low power, Harvard style, RISC microcontroller which provides 3-stage pipeline. It can manipulate operands in two address spaces: 3K-byte data RAM and 16K-word code ROM respectively. The RAM is further divided into 2048 X-memory and 1024 Y-memory for 16-bit co-

processor operations. The data of DHMM models and vector-quantized code book are saved in external flash memory and loaded to RAM whenever needed. The compressed voice prompts are stored in the flash or masked into data ROM. The ADC is Sigma-Delta type 256X over sample ADC with 14-bit resolution. The DAC provides a voice output channel with 8-bit resolution. As in figure 1, there are six 8-bit I/O ports and one serial port. An LCD controller controls LED lights. Timers, Battery Level Detection and Watch Dog Timer are also included.

The input signal is pre-amplified and filtered by a 300~3400Hz band-pass filter before sampled by ADC with 8 KHz sampling frequency. Then digital speech signal is analyzed and recognition carried out. Proper prompts is decoded and played after that. Also other outputs such as LCD are available.

3. SOFTWARE ARCHITECTURE

To handle the multi-task problem, a foreground / background software system with interrupts is designed. Apart from Reset and NMI, there are two interrupt vectors, each one have 8 interrupt sources. There are two 128-byte buffers preserved in RAM to alternately store the accumulated ADC data. Every 0.125 ms, data is sampled by ADC and saved to one buffer. After 8 ms, the full buffer is processed by the feature extraction function and the other buffer continues to store the ADC result. Thus two buffers are used in the alternation. Background tasks, such as calculating LPCC features and vector-quantizing each frame, are carried out when system is not engaged in sampling and accumulating autocorrelation values for the foreground tasks.

The 16-level hardware stack, which is provided by the system, is not enough for the task, so software stack is created and maintained by the program.

3.1 System Overview

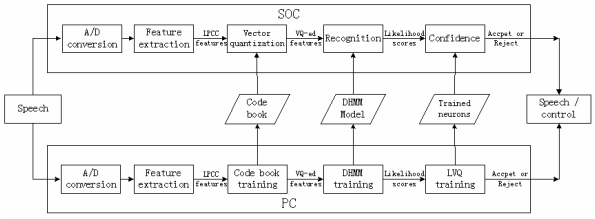

Fig.2 Software system overview

Speech features of LPCC and related features are used in the system. DHMM is employed to describe words that are to be recognized. There are two phases to run the system, one is the training phase which is performed on a PC to get the VQ code book and the DHMM models; the other is the recognizing phase which is carried out on chip by using the well trained models written on flash.

During the training phase, all the training utterances are processed by the front end to get frame based LPCC features, and then the VQ code book is trained by using these features with k-mean clustering. Parameters of DHMM are trained by using the vector quantized features with Baum-Welch algorithm.

During the recognition phase, the microphone accepts the speech, and features are extracted and vector-quantized. Then the vector-quantized frame sequence is matched with each word model with Viterbi decoding algorithm. The output likelihood scores are post-processed by the classifying neural network and a label is given to indicate to reject or to accept the utterance. This process completes the recognizing phase.

The advantage of this structure is that different vocabulary can be retrained easily and algorithms such as corrective training are transparent to the on-chip model format. With the changeable flash on chip, different vocabulary can be easily reconfigured for new recognition applications.

3.2 Front End Feature Extraction and Feature Selection

This speech recognition system chip uses LPCC, its first order delta component (ΔLPCC) , energy (E), its first and second order delta component (ΔE and $\Delta\Delta$E) as input features. Though Mel-Frequency Cepstral Coefficient (MFCC) is more powerful and robust to noisy conditions, it is computational formidable. After these features are combined and weighted heuristically, they are vector-quantized to further reduce recognition complexity.

Endpoint detection algorithm is used to separate the utterances from environmental sound and burst noises. A very simple endpoint detection algorithm based on Zero-Crossing Rate (ZCR) and frame energy is adopted for computational requirements.

Cepstral Mean Subtraction (CMS) has been proved to be an effective method to eliminate the channel characteristic from the cepstral coefficients and to improve recognition robustness for the mismatch between training and recognizing environments. But the traditional exponential window CMS routine requires more than 2-second voice to get a stable cepstral mean estimation [5]. But isolated Mandarin words seldom last over 1.5 second. To get a robust cepstral mean value estimate for 1.5 second speech, we initialize the cepstral mean with average of the first few voicing frames, and then update it frame by frame with a decay coefficient α to make it converge faster. α can be chosen by

$$\alpha = \ln 2 / (T \cdot F_s) \qquad (1)$$

Where T is the time voice lasts, Fs is the frame sampling frequency, not the 8 KHz sampling frequency.

3.3 Vector Quantization

Due to computation cost and limited RAM resource, the input feature sequences are quantized by a vector quantization (VQ). Each VQ number is assigned with an output probability in each state of each word model, so the output probability computation is simplified for our very limited memory resource. Though this quantization step changes one-step optimization to two-step optimization, yet it is acceptable. This step consumes most of the in the recognition process because VQ index for each frame can be obtained only after every code book item is searched. This is essentially a brute force search. So proper code book size is a key factor for system responding time.

The Self Organizing Map (SOM) is a competing neural network with two consecutive steps of global learning and local learning. [6] So we tried the Euclidian distance, two dimensional grid SOM as our VQ code book structure. The SOM-based VQ method can achieve better performance by reducing the quantization error while keeping the original feature geometry.

And the code book size is half of that of k-mean clustering. The SOM-based VQ implementation can be carried out in real time.

The features are represented by 16-bit fixed point digit on chip. From our observation, we know that most of the features falls to half of the digit range. So after cepstral weighting, the features are scaled and round-off heuristically to make use of the full range of the fixed point feature representation so as to further scatter code book and improve recognition rate.

Table 1. VQ recognition rate table

Test Set	Code Book size	RecogRate (%)
No VQ	/	98.9
VQ by k-mean	256	94.5
VQ by SOM	128	94.1
VQ + Round Off	256	95.1

As Table 1 shows, VQ degrades the recognition rate significantly; SOM can reduce VQ code book size while keeping comparable recognition rate; and recognition rate is further improved by scaling and rounding-off features before VQ. So we choose VQ with rounding-off as the final approach.

3.4 Discrete Hidden Marcov Model

Simple DHMM for whole word is chosen in our approach due to complexity requirements. [7] provides an overview on HMM. In our approach, each model contains several states which can only jump forward no more than one state. For two-syllable Mandarin word, 9 states are chosen heuristically by experiments. In training phase, state transition matrix is trained. In recognition phase, Viterbi search gives each model a logarithm likelihood score (LLS) and the model with maximum LLS is recognized as the correct result. A one-state jumpable silence model is added at both the start and end of the word model to absorb the non-voicing frames. This silence model is trained with non-voicing frame features.

As Table 2 shows, when hand labeled endpoint is used, DHMM with silence model (DWS) is as good as DHMM without silence model (DNS); when no endpoint detection algorithm is used, DWS significantly outperforms DNS; when simple endpoint detection algorithm is added, recognition rate remains the same. So the silence model, along with the simple ZCR and energy based endpoint detection algorithm, is selected to save both time and power consumption.

Table 2. Recognition rate table with silence model

RecogRate (%)	Hand labeled endpoint	No endpoint	Simple endpoint
DWS	95.1	94.9	95.0
DNS	95.1	92.4	92.9

3.5 Corrective Training

DHMM discussed above does not consider the inter-model interference and trains each word model separately. So if confusable words are in the same vocabulary, errors are prone to occur. To overcome this problem, corrective training introduced in [8] is adopted. This algorithm uses a confusable table to indicate the confusable words and re-estimated the confusable word's state probabilities through utilizing differences between LLS of the corresponding utterances. This procedure is iterated several times until it converges.

In our implementation, two modifications are made as follows. Firstly, the states can not be exactly aligned across models since sub-word model is not used. So we only adjust the true word's all state probability by using logarithmical likelihood ratio (LLR), which is is same for each state and defined as

$$LLR_{1,2} = LLS_1 - LLS_2 \tag{2}$$

Secondly, the confusable table is dynamically regenerated according to the confusion matrix of all the words after the iteration. This helps to adjust only the most confusable word in the iteration and thus improves recognition rate. Table 3 shows the recognition improvement.

Table 3. Corrective training recognition rate table

Corrective training method (iteration num)	RecogRate (%)
No Corrective training	95.1
Corrective training (1)	95.7
Dynamic confusable table(1)	95.9
Corrective training (5)	96.6
Dynamic confusable table(5)	97.4

Minimal Classification Error (MCE) criterion is also widely used to replace the Maximum Likelihood (ML) criterion [9]. Because of its continuous derivable form, MCE is more proper for CHMM and SCHMM (Semi-Continuous HMM). And we adopt the simple corrective training method for DHMM.

3.6 Confidence measure with LVQ Classifier

For our system, a simple confidence measure based on LLS is useful. Similar to [10], we extract a set of statistical features from the recognizing LLS and then use LVQ3 algorithm to train a classifier for INV (In Vocabulary) and OOV words.

Because busty noise and long utterance can be rejected by the endpoint detection algorithm, we take only 30 INV words and 30 OOV words of similar length, each of 500 utterance in the data set, and recognize them to obtain LLS for each model and label these utterances with 'INV' and 'OOV' respectively. LLS of each utterance are sorted in descending order as S_i, $i = 1, \ldots, N$, for the feature extraction phase.

The features we used include:

$$X_1 = S_1 / m_1 \tag{3}$$

$$X_2 = S_2 / S_1 \tag{4}$$

$$X_3 = | S_1 - E(\sum_{k \neq 1, S_k \geq \alpha S_1} S_k) | / S_1 \tag{5}$$

$$X_4 = S_{\min} / S_1 \tag{6}$$

in which m stands for mean LLS of this model, E stands for Expectation of LLS, α is the constant which is set by the user and ranging from 0.5 to 0.9, and $S_{\min}$ stands for the minimal LLS of an utterance. X_1, X_2 and X_3 are used in [10]. X_2 is a good indicator when there are similar words in the vocabulary. X_4 is good when OOV word is very dissimilar to INV word. Under this circumstance, both S_1 and S_2 can not exactly model the utterance. So X_4 provides an alternative here. A problem with X_3

is that it can only be determined after α is heuristically set, but the performance is unstable with different α. Then LVQ3 can draw the segmental linear classifier, in which 16 neurons are used. The detailed algorithm can be found in [6]. Also speech utterances of 80 persons are used for training, and the left of 20 persons for testing. When X_1 & X_2 & X_4 are used, we have obtained a result of 8.3% Rejection Rate and 4.5% False alarm Rate, which is not bad for many applications.

4. EVALUATION EXPERIMENTS

The system uses 23 dimensional LPCC and energy related features as input, and the code book with 128 vectors is used to quantize the features, then DHMM are trained to model the words that are to be recognized. The output likelihood scores are post-processed to provide a confidence measure to reject OOV words.

We sum up here the overall recognition rate boost during the steps in recognition error rate reduction. The final rates are all obtained in real environments, using on chip recognition directly.

Table 4. Overall recognition rate boost table

System description	RecogRate (%)
Base line	89.2
Cepstral Mean Subtraction	93.7
Vector Quantization	94.7
Silence model	95.4
Corrective training	96.7

Although the system recognition rate seems to drop at the Cepstral Mean Subtraction and Silence model step, it outperforms the base line system in the real environment test. Of course, the Corrective training step steadily boosts the system performance. And the Vector Quantization step is crucial for the real-time system implementation, though it will significantly increase the error rate.

Table 5. Different vocabulary recognition rate table

Vocabulary size	RecogRate (%)	RecogTime (real-time)
10	98.9	0.53
30	96.7	0.60
60	92.1	0.71

Different vocabulary tests in real environment are also conducted and shown in Table 5. The recognition time does not change much because VQ is the part that consumes most of the time. We can see that the this speech recognition chip can not deal with larger vocabulary. One probably explanation to this is that 7-bit VQ code book may become too imprecise for the HMM observation probability distribution, when compared with the original CHMM.

5. CONCLUSION

This discrete word, speaker independent, small vocabulary speech recognition system is designed for and implemented on an 8-bit MCU core. The system adopts LPCC and related features followed by a VQ Step as front end, and then the speech input is recognized with DHMM by using Viterbi decoding algorithm. The output LLS is further post-processed using LVQ to indicate whether the word is in the vocabulary or not. Experiments have shown that this system is designed with compromising goals of usability, flexibility, accuracy, speed and robustness. Also CVSD algorithm is used to achieve a user friendly speech in speech out interface.

The system can be used for toys, office devices, and other consumer electronic products for it can offer an accurate, flexible and robust solution with extremely low cost.

The key characteristics of the chip are summarized in table 6.

Table 6. Characteristics of the chip

Process technology	0.5 um CMOS
Package	100-pin QFP
Clock frequency	8.2MHz
Supply voltage	2.2V~5.25V
Power consumption	60mw
Recognition speed	0.60 real-time
Recognition rate	96.7% (30 phrases)

6. ACKNOWLEDGMENT

The research is supported by the National Natural Science Foundation of China (NSFC No. 60272016). Thanks go to Li Xiaoyu，Liang Weiqian, and Dong Ming. And special thanks go to Yu Yang.

7. REFERENCES

[1] Bi N, et al. A robust speech recognition system embedded in CDMA cellular phone chipsets. *Proc ICASSP'02,* pp. 3804-3807, April, 2002.

[2] *http://www.sensoryinc.com*. RSC-4x Data Sheet.

[3] Yuanyuan Shi, Jia Liu and Rensheng Liu, Single-Chip Speech Recognition System Based on 8051 Microcontroller Core, *IEEE Trans. on Consumer Electronics*, vol. 47, No.1, pp. 149-154, 2001

[4] *S3CB519/FB519 user manual*, Samsung Electronics Co., Ltd. 2002.

[5] Xuedong Huang, et al, *Spoken Language Processing,* Prentice Hall, New Jersey, 2001.

[6] T. Kohonen, "The Self-Organizing Map" *Proc. IEEE,* Vol. 78, No.9, September 1990.

[7] L. Rabiner, B-H. Juang, *"Fundamentals of Speech Recognition",* Prentice Hall, 1993.

[8] L. R. Bahl, P. F. Brown, et al. "Estimating Hidden Markov Model Parameters So As To Maximize Speech Recognition Accuracy" *IEEE Trans. Speech and Audio Processing,* Vol. 1, No.1, January 1993.

[9] B-H. Juang, W. Chou, C-H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition" *IEEE Trans. Speech and Audio Processing,* Vol. 5, No.3, May 1997.

[10] Yonggang Deng, et al, "Speech Recognition Algorithm and implementation on Palm PC", *Research and Development of Computers*, vol. 37, No.8, August 2000. (In Chinese)