Automatic Speech Recognition

JIA Pei

Email: jp4work@gmail.com

Vision Open Working Group

First Edition

April 22, 2010

Contents

Abstract

ASR has already been researched and applied as a HCI for over thirty years [6]. Normally, in order to apply speech recognition, an entire system needs to be developed from scratch. Fortunately, several famous open source speech recognition systems are available on the Internet, such as HTK developed by Cambridge University [38, 39] (note: HTK has license restriction), Julius maintained by Kyoto University [16], ISIP speech recognition environment by Mississippi State University [7, 23] and CMU Sphinx [18, 30]. Modern general-purpose speech recognition systems, including all the above four open source ASR systems, are generally based on Mel Frequency Cepstral Coefficients (MFCCs) for audio signal presentation and Hidden Morkov Models (HMMs) to model the speech stochastic process.

HMM was first described in a series of statistical papers by Baum [3] and some other authors in the 1960's. Speach recognition reseach based on HMM started in the mid-1970s. A representative ASR at that time is Dragon [1, 2]. After that, variants of HMM were explored, such as discrete HMMs [19], semicontinuous HMMs [13–15] and continuous HMMs [24]. In 1989, Rabiner reviewed HMM theories and summarized some problems in speach recognition that could be modeled by HMM [25]. All the above HMM related researches are summarized in the online tutorial "Ten years of HMMs" [5].

In fact, from the viewpoint of pattern recognition and machine learning, HMM, as well as another famous control model - Kalman filter could be viewed as examples of Dynamic Bayesian Networks (DBNs) [10]. Murphy reviewed this point in his PhD thesis [21] and stated HMMs has limited "expressive power" but DBNs generalize HMMs by allowing the state space to be represented in factored form, rather than a single discreted random variable. Therefore, DBNs, also known as directed graphic models [4] should be a more generalized tool for speech recognition. In fact, early in 1995, Zweig has already put DBNs into speech recognition in his PhD thesis [40].

The entire architecture of CMU Sphinx4 speech recognition system is cited in figure 1.

In the following section, how to generally construct an entire ASR system is elaborated.

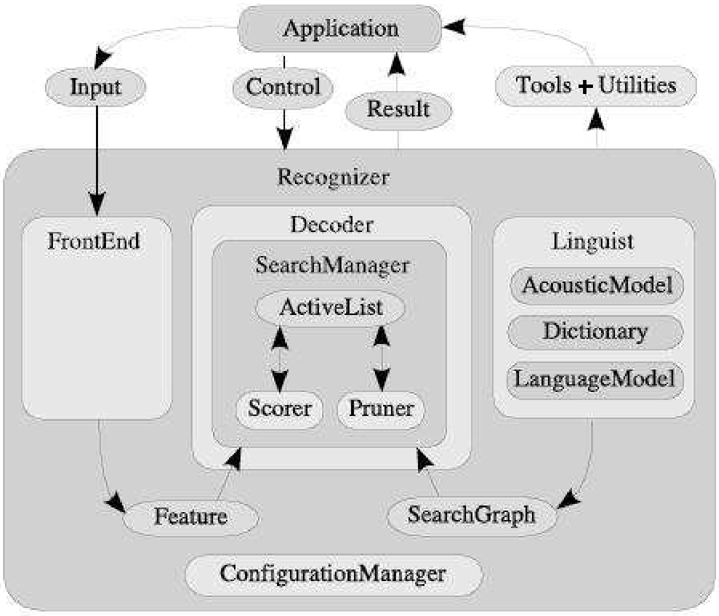


Figure 1: Sphinx4 ASR Framework [30]

0.1 Model the Speech Stochastic Process

0.1.1 Discretize the Speech Signal

Substantially, speech can be looked on as a continuous stochastic process. Operably, it is dealt with as a discrete stochastic process by sampling first and then windowing. But how to discretize the speech signal?

We first denote the sample rate as S_r;

- The frequency range of the sound that human being can hear is between 20 to 20,000Hz. According to Nyquist theory, in order to be able to analyze the sound (1D signal) which can be heard, a sample rate of at least 2*20,000 is necessary. The most common sample rate for speech/music is $S_r = 44,100$Hz, which absolutely satisfies Nyquist theory.

- Refer to [33], the frequency of ordinary human speech varies between 512Hz and 2,048Hz, between 2,048Hz and 8,192Hz for labial and fricative sounds, and from 8,192Hz to 11,000Hz for the sound of letter "S". Therefore, according to Nyquist theory, the sample rate could be just over 11,000*2=22,000Hz. That's why $S_r = 22,050$Hz is also often used in audio files. In fact, in the above speech recognition OSSs, S_r is often selected as 8,000Hz, 16,000Hz and 32,000Hz.

- In order to analyze the frequency spectrum of a signal, at least one entire period of the specific signal harmonic component should be covered. The biggest period for the harmonic component from the sound that can be heard is calculated as 1/20ms, which means if harmonic component with the lowest hearable frequency needs to be covered in one sample window, the window size should be at least of length $S_r/20$.
- From [37], "the voiced speech of a typical adult male will have a fundamental frequency of from 85 to 155 Hz, and that of a typical adult female from 165 to 255 Hz". Therefore, although we can hear a sound below 85 Hz, we are never able to speak out such a sound. In order to be able to analyze the harmonic component with the lowest frequency of the voice that can be spoken out by human being, the window size should be at least of length $S_r/85$.
- The window size should be narrow enough that the speech articulators do not significantly change in that window.
- In order to ease later Fourier transform, it's better to ensure the window size be $2^N, \quad N = 1, 2, \cdots$

In sum, we give out the following conclusion:

- Sample rate: 44,100Hz, 22,050Hz, 32,000Hz, 16,000Hz, 8,000Hz.
- Suitable window size to discretize the speech signal: 128, 256, 512, 1,024 and 2,048.

After this speech discretization, it's easy for us to understand why the adjacent frames (windows) are partially overlapped. A Hamming window operating on the speech signal attenuates the signal at both window edges, which causes the signal at the edge is not well analyzed in this frame. Therefore, we may recenter the sample window in the next frame partially overlapping with the current sample window, in order to analyze the entire speech signal without loss.

0.1.2 Frontend

Several canonical feature extraction methods have been proposed for speech recognition as the frontend, including MFCC, LPC, RASTA-PLP [11, 12] and TECC [8], etc. Variants based on the above audio processing methods flooded by removing preemphasis, adding liftering technology, weighting filter banks rather than the triangular way, etc. Here, we only explain MFCC in detail.

Preprocessing

We first denote the raw wave data as $s_n, n = 0, 1, \cdots$. The audio signal preprocessing is summarized in the following several steps:

1. Preprocess the signal by holistic preemphasis, namely, by applying the first order difference equation

$$s'_n = s_{n+1} - ks_n \qquad n = 0, 1, \cdots \tag{1}$$

   to all the samples s_n. Here k is the preemphasis coefficient which should be in the range $0 \leq k < 1$. In our application, $k = 0.97$. Refer to figure 3.
   Note: Preemphasis changes the voice frequency characteristics.

2. Afterwards, a window function is applied to observe a single frame of N samples and extract features from the corresponding local samples. It is usually beneficial to taper the samples in each window so that discontinuities at the window edges are attenuated. This could be done by applying Hamming window functions to the samples after preemphasis s'_n in each window as follows:

$$s''_n = 0.53836 - 0.46164 \cos\left(\frac{2\pi n}{N-1}\right) s'_n \qquad n = 0 \cdots N-1 \tag{2}$$

   Hamming window is visually described in figure 2.

The audio signal preprocessing could be summarized in figure 3.

MFCC

After preprocessing, MFCCs are calculated in the following substeps.

1. Carry out the Discrete Fourier transform (DFT) (Refer to [34] on (a windowed excerpt of) a signal, here $s''_n, n = 0 \cdots N-1$, the frequency spectrum S could be obtained by

$$S_k = \sum_{n=0}^{N-1} s''_n e^{-2\pi ikn/N} \qquad k = 0, \ldots, N-1 \tag{3}$$

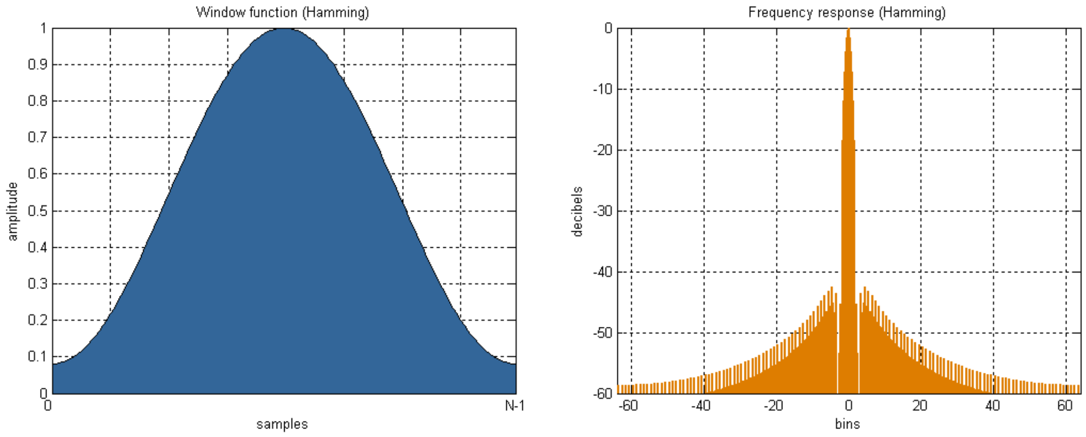


Figure 2: Hamming Window Function [32]

where k corresponds to a frequency f_k. The sampling theory tells us that

$$f_k = kf_s/N \tag{4}$$

where f_s is the sampling frequency in Herz [26].

2. Build up the triangular overlapping filter banks as shown in figure 5 in terms of the evenly distributed mel scale corresponding to the frequency values as in figure 4, where mel scale is defined as:

$$m = 1127.01048\ln(1 + f/700) \tag{5}$$

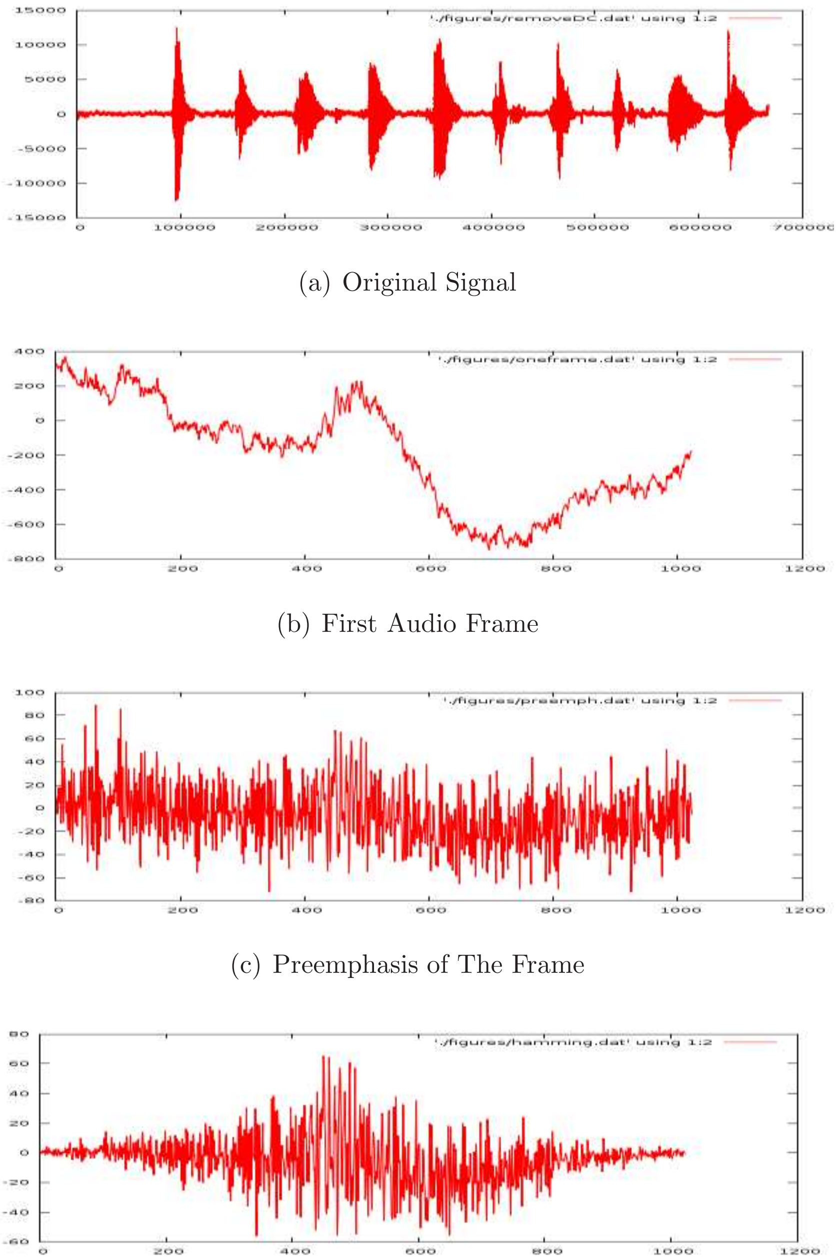
With the M built triangular overlapping filter banks $b_u, u = 1, 2, \cdots, M$ which are unevenly distributed in FFT frequency with $M+2$ frequency endpoints $p_u, u = 0, 1, 2, \cdots, M+1$, the M mel bins could be computed as:

$$B_u = \sum_{k=0}^{N-1} \|S_k\| b_u(f_k) \qquad u = 1, 2\cdots, M \tag{6}$$

where

$$b_u(f_k) = \begin{cases} 0 & \text{if } f_k < p_{u-1} \text{ or } f_k > p_{u+1} \\ \frac{f_k - p_{u-1}}{p_u - p_{u-1}} & \text{if } p_{u-1} \leq f_k < p_u \\ \frac{p_{u+1} - f_k}{p_{u+1} - p_u} & \text{if } p_u \leq f_k \leq p_{u+1} \end{cases} \qquad u = 1, 2, \cdots, M \tag{7}$$

If the concerned frequency is limited between 0 and 8,000, and 19 filters are finally selected to compose the filter banks, namely, $p_0 = 0$,



(a) Original Signal

(b) First Audio Frame

(c) Preemphasis of The Frame

(d) Convoluted by Hamming Window

Figure 3: Speech Signal Preprocessing

$p_{M+1} = 8,000$ and $M = 19$, the frequency edges of triangular filter banks are easily calculated as the following vector **p**=(0.0, 94.0, 200.6, 321.6, 458.7, 614.3, 790.8, 991.0, 1218.1, 1475.6, 1767.8, 2099.2, 2475.1, 2901.4, 3385.0, 3933.6, 4555.8, 5261.5, 6062.0, 6970.0, 8000.0), as shown in figure 5.

It's reported in [20] that "for speech/music this classification problem, the results are (statistically) significantly better if Mel-based cepstral features rather than linear-based cepstral features are used." Here in our application, we just take this conclusion as true.

3. Take a *log* function on the above obtained M mel bins.

$$B'_u = \ln(B_u) \qquad u = 1, 2 \cdots, M \tag{8}$$

4. Take the Discrete Cosine Transform (DCT) [17] of the list of mel bins, as if it were a signal.

$$C(u) = \sum_{u=1}^{M} B'_u \cos\left[\frac{\pi(2m-1)u}{2M}\right] \qquad u = 1, 2 \cdots, M \tag{9}$$

   The MFCCs are just the amplitudes of the above obtained spectrum $C(u)$, which will be used as the extracted features for both training and recognition.

The entire MFCC process could be summarized in figure 6.

0.1.3 Analysis of MFCC

In order to demonstrate the performance of MFCC representation, a segment of speech is used to try out its reconstruction capability. Based on white noise modulation, this speech segment can be roughly reconstructed, which shows that MFCC are good features capable to represent the speech. Actually, key features used for recognition might not be the features which can reconstruct the original speech. But here, we only investigate the reconstruction capability. By the way, TECC is recently reported to be an even better front end for audio presentation (refer to [8]).

During the whole process of MFCC, all steps are invertible except two:

1. The absolute values rather than real and imaginary parts of DFT coefficients are used. So, to reconstruct the speech, white noises are randomly generated and amplitude-modulated (AM) by the DFT absolute values. The original signals are guessed through this way without too much distortion. (Refer to 0.1.3)

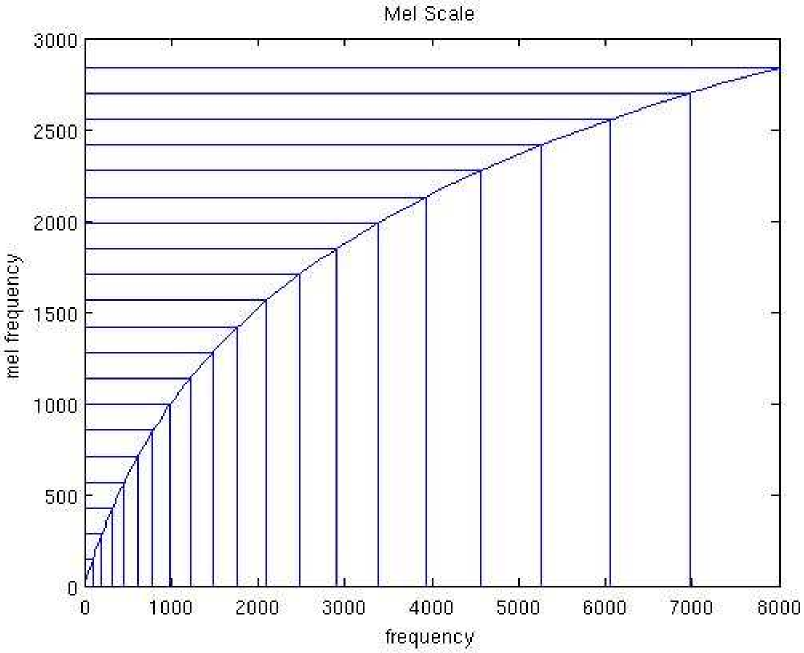


Figure 4: Mel Hz Plot with Even Distributed Mel Slots

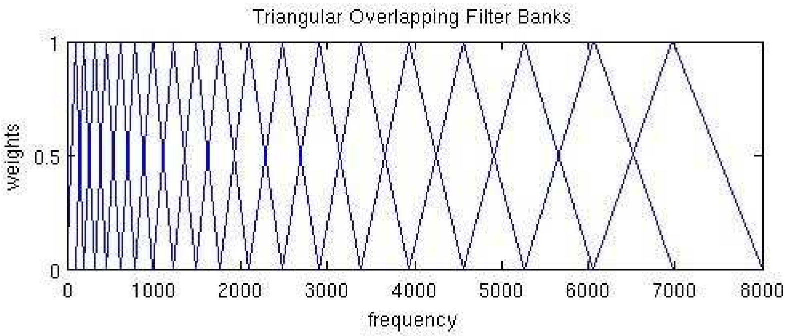
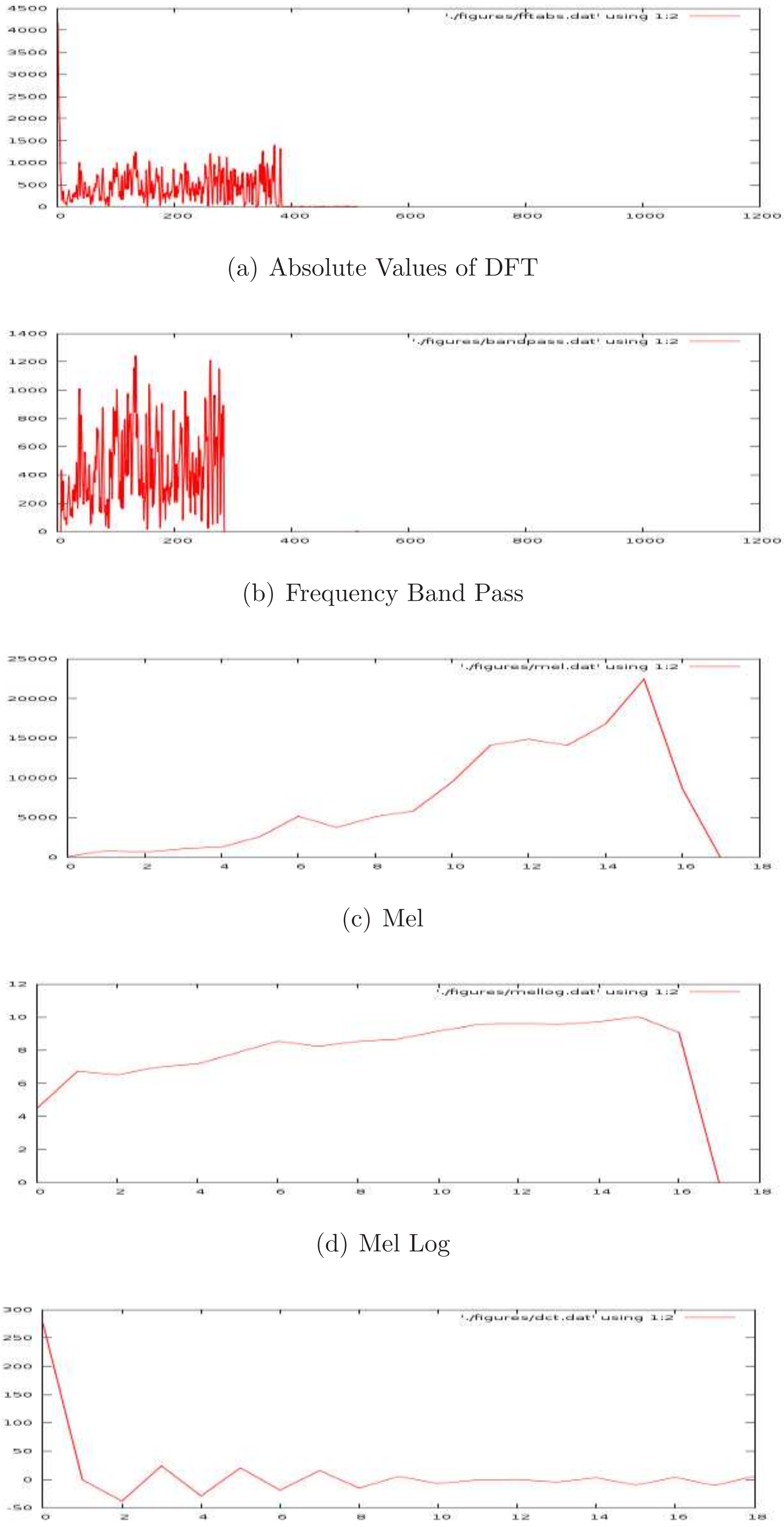


Figure 5: Filter Banks



(a) Absolute Values of DFT

(b) Frequency Band Pass

(c) Mel

(d) Mel Log

(e) DCT

Figure 6: MFCC

2. Frequency to mel scale mapping is not 1 to 1. Generally, MFCC is an energy compaction method, which tries to pack energies from lots of frequency bins to a small number of mel bins. During the process of voice reconstruction, a pseudo inverse is calculated so the energy from a small number of mel bins can be roughly redistributed to a great number of frequency bins.

White Noise Modulation

With known $\|S_k\|, k = 0, 1 \cdots N-1$ only (refer to (6)) and without any prior knowledge about the signal itself, recovering s''_n in (3) is difficult. One possible way is to randomly generate Gaussian white noises of mean 0 and variance 1, namely, a Gaussian white noise sequence $w_n, n = 0, 1 \cdots N-1$. Every number w_n is a random number generated from the normal distribution $\mathbf{N}(0, 1)$. Hopefully, w_n will not change the energy of the original signal.

Afterwards, DFT is carried out on this randomly generated Gaussian white noise sequence.

$$W_k = \sum_{n=0}^{N-1} w_n e^{-2\pi i k n/N} \qquad k = 0, \ldots, N-1 \tag{10}$$

Finally, the generated real and imaginary parts from W_k are amplitude-modulated (AM) by the sequence $\|S_k\|$.

Redistribute Energy from Mel Bins to Frequency Bins

In fact, (6) is just a matrix multiplication, where $\|S_k\|$ is a column vector of length N, B_u is a column vector of length M, and $b_u(f_k)$ is a matrix of size $M * N$, which reflects the mapping relationship from frequency bins to mel bins. From (7) and (4), if the number of mel bins M, the size of the sample window N, and the sample rate f_s are known values, this mapping matrix $b_u(f_k)$ can be determined preliminarily, then, its pseudo inverse can be directly used to calculate $\|S_k\|$ from B_u. Figure 7 just shows all the weights that have been adopted in our frequecy to mel mapping.

Voice Reconstruction

Figure 8 just shows the entire process of recoverying a single audio frame from DCT parameters only.

As we can see, although (c) in figure 8 is quite similar to (b) in figure 6, there are big differences between (d) in figure 8 and (d) in figure 3. That means, for one audio frame, the signal can't be well reconstructed. However,

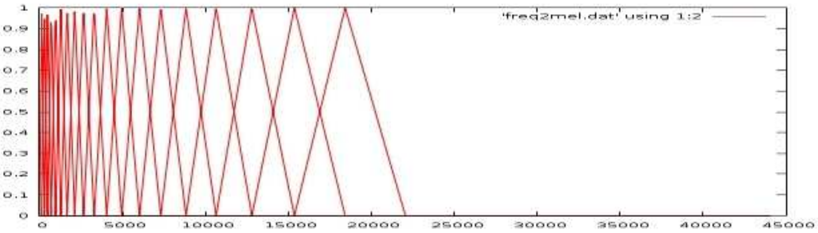


Figure 7: Used Filter Bank Weights

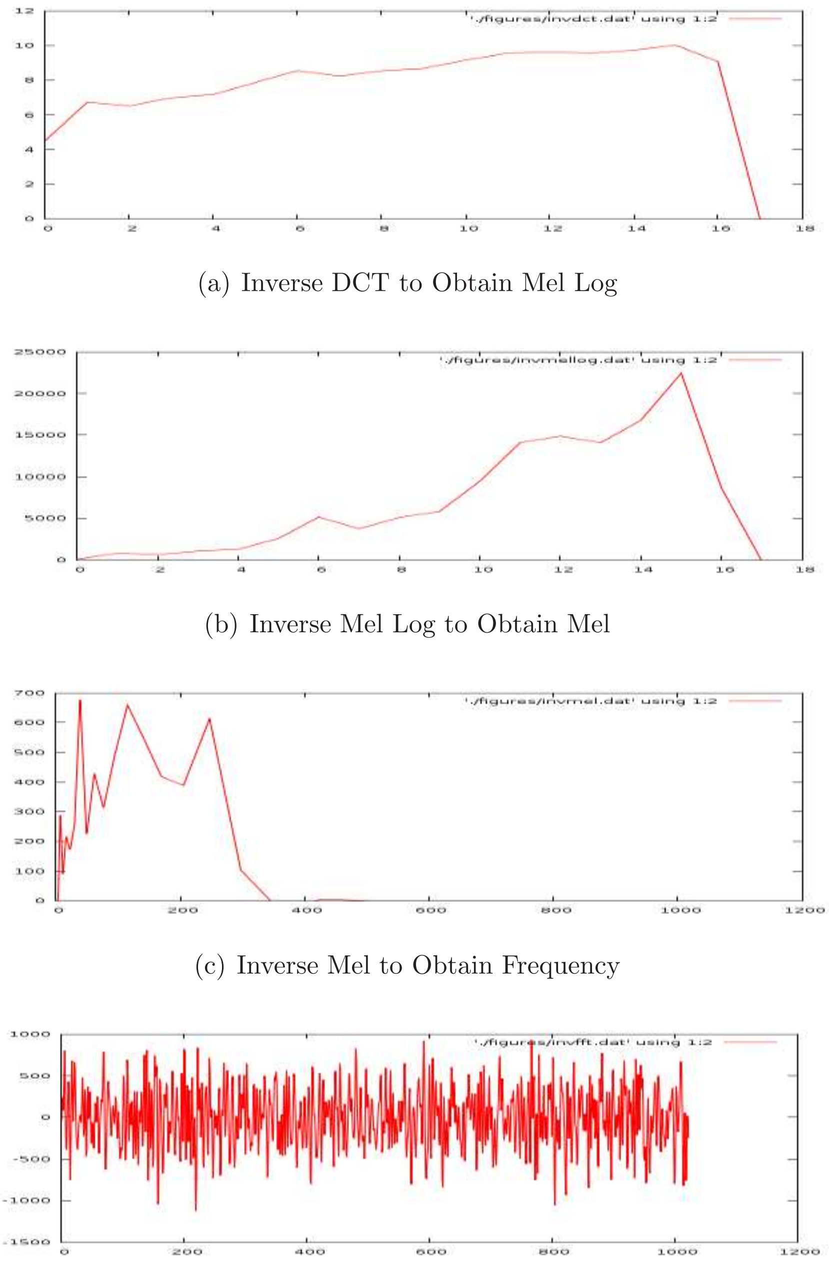
comparing (g) or (h) in figure 8 and (a) in figure 3, we may conclude that from a long-term aspect of view, the signal can be reconstructed reasonably well, without considering the amplitude infidelity.

Two methods are finally used to join the recovered audio frames into a whole signal. The first one (refer to (g) in figure 8) ignores the overlapped part of every audio frames, the second one (refer to (h) in figure 8) averages the overlapped part of every two neighbour frames. Experiments show the first method gives better results.

0.1.4 Acoustic Model

As mentioned in 0.1.1, the window size should be short enough to ensure there is only one single sound pronounced within this specific window. In fact, all sounds of all Latin languages could be represented in the standard IPA, short for International Phonetic Alphabet [35]. IPA is just such a system to represent distinctive phonemes, intonation, and the separation of words and syllables in spoken language. As of 2008, IPA defines 107 distinct letters, 52 diacritics, and 4 prosody marks to represent the spoken languages.

Apparently, if every sound unit is looked on as one of the hidden states, and every vector of MFCCs is looked on as one possible observation state, the entire speech process could just be looked on as a Hidden Markov Model (HMM). Why is the model a HMM rather than just a Markov Model is because the observation states can't be unambiguously mapped to the corresponding hidden states, namely, the observation states and the hidden states are not 1-1 map. Acoustic models contain a statistical representation of the distinct sound units that make up a whole word in the dictionary. Each distinct sound unit corresponds to a phoneme.

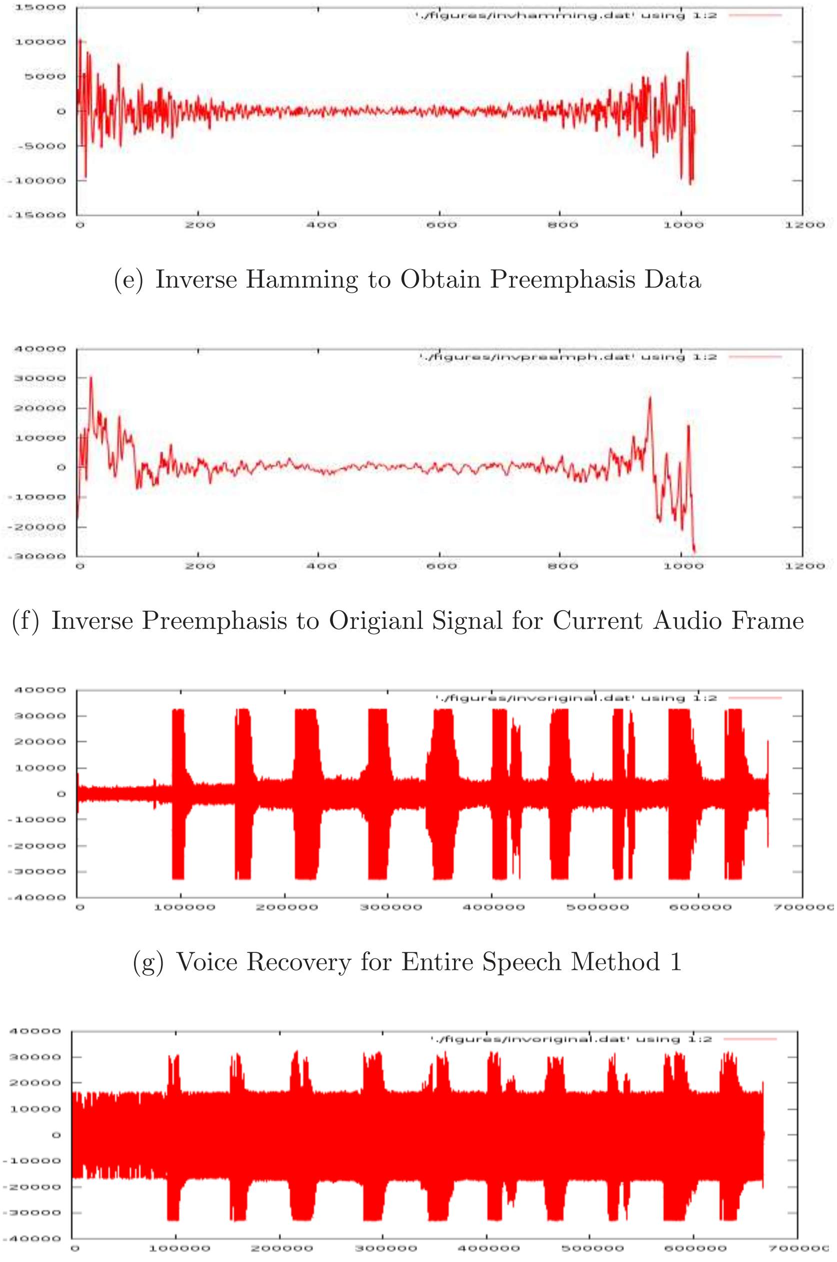


(a) Inverse DCT to Obtain Mel Log

(b) Inverse Mel Log to Obtain Mel

(c) Inverse Mel to Obtain Frequency

(d) Inverse Frequency to Obtain Signals after Preprocessing, based on White Noisy Modulation



(e) Inverse Hamming to Obtain Preemphasis Data

(f) Inverse Preemphasis to Origianl Signal for Current Audio Frame

(g) Voice Recovery for Entire Speech Method 1

(h) Voice Recovery for Entire Speech Method 2

Figure 8: Voice Reconstruction

0.1.5 Language Model

Language models contain a list of words and their probability of occurrence in a given sequence. Unlike acoustic models which provide phoneme-level speech structure, language models provide word-level language structure. Language models typically fall into two categories: graph-driven models [28, 29] and N-Gram models [27,31]. Graph-driven models could be looked on as 1D Markov model, which means the word probability is only based on the previous word; while N-Gram models could be looked on as $(n-1)$D Markov model, which means the word probability is estimated by the previous $n-1$ words.

0.1.6 Decoder

To recognize all words of a sentence in context is obviously a much more complicated task than to just recognize isolated words. If we simply consider to recognize the isolated words, then, decoder could be looked on as a search engine, which essentially look for the most likely single word in the dictionary that has the most similar pronunciation as what the user has pronounced.

Therefore, speech recongnition finally becomes a graph search problem to find the most likely phoneme sequence. Generally speaking, there are many categories of search algorithms, such as (refer to [36]): brute-force search or exhaustive search, heuristic search including A^* search, breath-first search, depth-first search, etc. Since our acoustic model is based on HMM, Viterbi search algorithm specific for HMM will be adopted in our experiments.

0.2 Experimental Results

We carry out our experiments using CMU with the following testing configuration parameters:

"WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz.Model" is used as the acoustic model which is configured in the XML file before execution. A simple explanation of this acoustic model is as follows:

- WSJ – Wall Street Journal [9], which refers to a large speech data set (or "corpus") that was suitable for training the acoustic model. WSJ is read by many adult male and female speakers with American English. The dataset is available online in LDC (Linguistic Data Consortium);

- 8gau – An HMM models a process using a sequence of states. Associated with each state, there is a PDF (Probability Density Function). A popular choice for this function is a Gaussian mixture, that is, a summation of Gaussians. A single Gaussian is defined by a mean and a variance, or, in the case of a multidimensional Gaussian, by a mean vector and a covariance matrix, or, under some simplifying assumptions, a variance vector. Here, a mixture of 8 Gaussian distributions is used in this acoustic model;

- 13dCep – A Cepstrum is the result of taking the Fourier transform (FT) of the decibel spectrum as if it were a signal. Its name was derived by reversing the first four letters of "spectrum". In Sphinx4, 13 Cepstral, 13 ∇Cepstral (first order derivative) and 13 ∇^2Cepstral (second order derivative) coefficients were used to model the speech spectra (Please refer to [22] for how Sphinx4 apply time-derivative parameters). Thus, there should be 39 Cepstrum parameters in total;

- 16k – the training speech files are sampled at the rate of 16kHz;

- 40mel – 40 Mel scale filters are used to suit for 16k different audio frequencies. To convert f hertz into m mel, $m = 1127.01048log_e^{1+f/700}$; and the inverse, $f = 700(e^{m/1127.01048} - 1)$;

- 130Hz_6800Hz – Possible minimum and maximum audio frequencies for an ordinary human.

In our application, the IW acts according to the following 6 commands by the corresponding words: "forward", "backward", "left", "right", "start", "stop". Please refer to the online video at `http://www.visionopen.com/products/JP_IWCommands.html`

Bibliography

[1] J. K. Baker. The dragon system–an overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):24–29, February 1975. IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., USA.

[2] J. K. Baker. *Stochastic modeling as a means of automatic speech recognition.* PhD thesis, Carnegie Mellon University, 1975.

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist*, 41(1):164–171, 1970.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[5] O. Cappé. Ten years of hmms, March 12 2001.

[6] J. Clark and R. Roemer. Voice controlled wheelchair. *Arch. Physical Med. Rehab.*, 58:169–175, 1977.

[7] N. Deshmukh, A. Ganapathiraju, J. Hamaker, J. Picone, and M. Ordowski. A public domain speech-to-text system. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 2127–2130, September 1999.

[8] D. Dimitriadis, P. Maragos, and A. Potamianos. Auditory teager energy cepstrum coefcients for robust speech recognition. In *Proceedings of European Speech Processing Conference*, Lisbon, Protugal, September 2005.

[9] J. Garofalo, D. Graff, D. Paul, and D. Pallett. Csr-i (wsj0) complete. Linguistic Data Consortium, Philadelphia, 1993.

[10] Z. Ghahramani. *Adaptive Processing of Sequences and Data Structures . Lecture Notes in Artificial Intelligence.*, chapter Learning Dynamic Bayesian Networks, pages 168 – 197. Springer-Verlag, Berlin, 1998.

[11] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, April 1990.

[12] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transaction on Speech and Audio Process*, 2(4):578–589, October 1994.

[13] X. Huang. Semi-continuous hidden markov models for speech signals. *Computer Speech and Language*, 3(3), July 1989.

[14] X. Huang. Phoneme classification using semicontinuous hidden markov models. *IEEE Transactions on Signal Processing*, 40(5), May 1992.

[15] X. Huang, H. Hon, and M. Hwang. A comparative study of discrete, semicontinuous, and continuous hidden markov models. *Computer Speech and Language*, 7(4), October 1993.

[16] Julius. Multipurpose large vocabulary continuous speech recognition engine. Technical report, Kyoto University, December 2001. Translated from the original Julius-3.2-book by Ian Lane ?Kyoto University.

[17] Syed Ali Khayam. The discrete cosine transform (dct): Theory and application. Technical report, Department of Electrical & Computer Engineering, Michigan State University, March 10th 2003.

[18] P. Lamere, P. Kwok, W. Walker, E. Gouvea, R. Singh, B. Raj, and P. Wolf. Design of the cmu sphinx-4 decoder. Eurospeech, 2003 (Eurospeech 2003) TR2003-110, Mitsubishi Electric Research Laboratories (MERL), September 2003.

[19] K. F. Lee, H. W. Hon, and R. Reddy. An overview of the sphinx speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:35–45, January 1990. Research supported by DARPA.

[20] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of International Symposium on Music Information Retrieval (Music IR 2000)*, Plymouth, MA, USA, October 23-25 2000. University of Massachusetts at Amherst.

[21] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, July 2002.

[22] Y. Obuchi and R. M. Stern. Normalization of time-derivative parameters using histogram equalization, 2003.

[23] Joe Picone and the Staff at ISIP. *Fundamentals of Speech Recognition: A Tutorial Based on a Public Domain C++ Toolkit*. Institute for Signal and Information Processing (ISIP), Mississippi State University, August 15 2002.

[24] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer. The 1996 hub-4 sphinx-3 system. In *In Proceedings of the DARPA Speech Recognition Workshop*, Chantilly, VA, USA, February 1997.

[25] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE*, volume 77 of *2*, pages 257–286, Febrary 1989.

[26] S. Sigurdsson, K. B. Petersen, and T. Lehn-Schiler. Mel frequency cepstral coefcients: An evaluation of robustness of mp3 encoded music. In *7th International Conference on Music Information Retrieval (ISMIR 2006)*, Victoria, Canada, October 8 - 12 2006.

[27] C. Tillmann and F. Xia. A phrase-based unigram model for statistical machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL 2003)*, volume 2, pages 106–108, Edmonton, Canada, 2003. IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA.

[28] J.-P. Vert and M. Kanehisa. Graph-driven features extraction from microarray data using diffusion kernels and kernel cca, December 9-14 2002.

[29] J.-P. Vert and Y. Yamanishi. Supervised graph inference, December 13-18 2005.

[30] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx-4: A flexible open source framework for speech recognition. Technical Report SMLI TR2004-0811, SUN MICROSYSTEMS INC., November 2004.

[31] Wikipedia. N-gram — wikipedia, the free encyclopedia, 2007. [Online; accessed 28-December-2007].

[32] Wikipedia. Window function — wikipedia, the free encyclopedia, 2007. [Online; accessed 29-December-2007].

[33] Wikipedia. Audio frequency — wikipedia, the free encyclopedia, 2008. [Online; accessed 25-December-2008].

[34] Wikipedia. Discrete fourier transform — wikipedia, the free encyclopedia, 2008. [Online; accessed 13-December-2008].

[35] Wikipedia. International phonetic alphabet — wikipedia, the free encyclopedia, 2008. [Online; accessed 17-December-2008].

[36] Wikipedia. Search algorithm — wikipedia, the free encyclopedia, 2008. [Online; accessed 21-December-2008].

[37] Wikipedia. Voice frequency — wikipedia, the free encyclopedia, 2008. [Online; accessed 18-December-2008].

[38] S. Young. The htk hidden markov model toolkit: Design and philosophy. Technical Report Technical Report TR.153, Department of Engineering, Cambridge University, 1994.

[39] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book*. Cambridge University Engineering Department, Cambridge University Engineering Department, Cambridgeshire, UK, for htk version 3.4 edition, December 2006.

[40] Geoffrey G. Zweig. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, University of California, Berkeley, 1998.