# Chapter 3

# Feature Extraction

Based on the speech features introduced in chapter 2 *Speech Signal Features*, this chapter will present the approach to extract speech features. Feature exaction includes two primary steps - pre-processing and Mel-frequency cepstral coefficients, corresponding to *pre-process* block and *MFCC* block in Fig. 1.1 on page 9. Pre-processing involves section 3.1 *Analog-to-Digital Conversion* to section 3.4 *Thresholds* while MFCC extraction incorporates section 3.5 *Mel-frequency Cepstral Coefficients Extraction*.

## 3.1 Analog-to-Digital Conversion

Voices in real life are analog signals, hence before conducting digital signal processing techniques, analog-to-digital conversions are required.

Given a continuous-time signal $s(t)$, we define the sampled signal by

$$s[n] = s(nT_s) = s(\frac{n}{F_s}) \tag{3.1}$$

where $T_s$ is the sampling interval and $F_s$ is the sampling rate.

$T_s$ should be carefully chosen in order to avoid distortion caused by aliasing. Telephony since the 1950s limits the information bandwidth to 300-3400 Hz [10]. However, in normal conversational speech, the frequency content is mainly between 0-8000 Hz [11]. According to Nyquist-Shannon sampling theorem, we set the folding frequency $\frac{F_s}{2} = 8000$ Hz, i.e. $F_s = 16$ kHz.

## 3.2 Pre-emphasis

The speech production system naturally attenuates voiced speech by approximately 20 dB per decade [12]. In addition, human hearing is more sensitive to frequency band from 1 kHz - 5 kHz [12]. However, as is depicted in Fig. 3.2, frequency components below 1 kHz predominantly comprise the spectrum. Hence, it is advantageous to employ a high-pass filter to amplify the high frequency range.

A first-order FIR filter represented by (3.2) is widely implemented, including the previous group where $\alpha = 0.95$ [10]. The merits of this FIR filter include simplicity and efficiency. However, Fig. 3.1 (orange dash-dot line) shows that frequencies below 500 Hz are severely suppressed even though frequencies above 3 kHz are successfully amplified. Considering the potential interference caused by high-frequency noise, attenuating low frequencies too much will result in the decline of signal-to-noise ratio (SNR).

$$y[n] = x[n] - \alpha x[n-1] \tag{3.2}$$

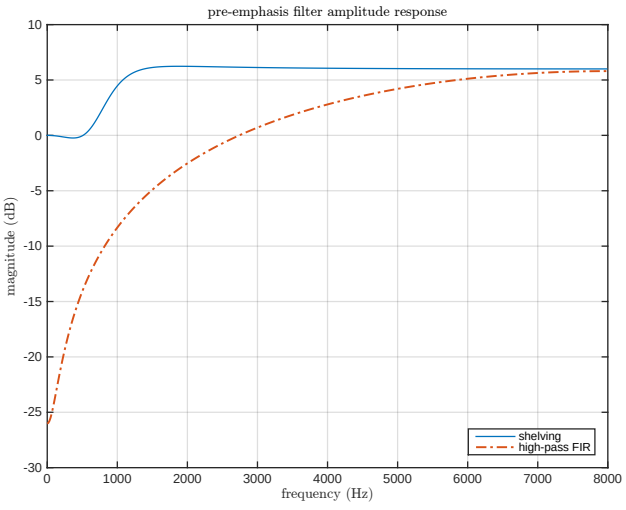


Figure 3.1: Pre-emphasis Filters

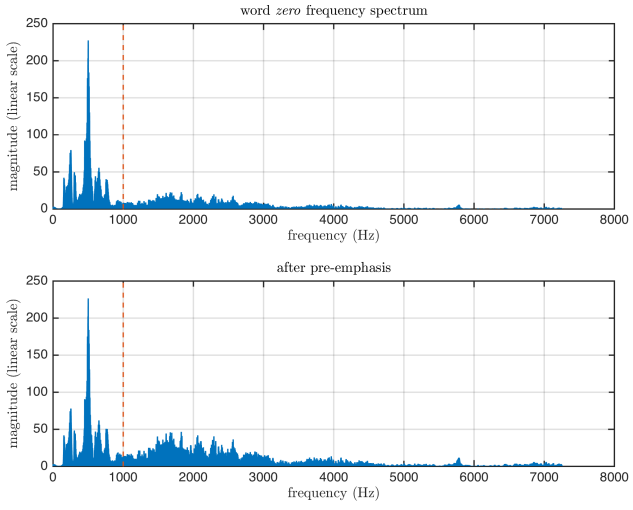


Figure 3.2: Spectra of Word *zero*

Suggested by Professor Erik Weyer, we try to devise a second-order *shelving filter* widely utilized in audio equalization to pre-emphasize the speech signal. With the aid of [13], setting center frequency of transition band $F_c = 1000$ Hz and gain $G = 6$ dB, eventually we obtain an appropriate filter that effectively amplifies high-frequency components without attenuating

low frequencies. Fig. 3.1 (navy solid line) shows the frequency response of this shelving filter and Fig. 3.2 shows the spectrum and pre-emphasized spectrum of word *zero*.

$$y[n] = \frac{1}{a_0}\Big(b_0x[n] + b_1x[n-1] + b_2x[n-3] - a_1y[n-1] - a_2y[n-2]\Big) \tag{3.3}$$

where

$$\begin{cases} a_0 = 1 \\ a_1 = -1.523796 \\ a_2 = 0.649345 \end{cases} \qquad \begin{cases} b_0 = 1.861856 \\ b_1 = -3.102851 \\ b_2 = 1.366544 \end{cases} \tag{3.4}$$

The pre-emphasis filter is mathematically represented by (3.3) and (3.4). Equations for calculating filter coefficients $a_i$ and $b_i$ $(i = 0, 1, 2)$ are available in the Appendix on page 96.

## 3.3 Framing & Windowing

Speech signals are time-varying signals, but due to the inertial motion of articulators (speech organs such as the tongue, lips and palate), speech can be considered statistically stationary in a short-time period (approximately 30 ms) [14]. The time period of 30 ms indicates 30 ms $\times$ 16000 Hz = 480 samples per frame. We take $N = 512$ to achieve a power of 2 in order that Fast Fourier Transform can be effectively conducted during the following *power spectrum* procedure in section 3.5 *Mel-frequency Cepstral Coefficients Extraction*.

The framing operation can be finished by multiplying the signal by a moving window. For the $j$-th frame and frame length $N$, mathematical equation is given in (3.5).

$$s_j[n] = \begin{cases} w[n]s[n + jN] & n = 1, 2, \ldots, N \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

The simplest and easiest-to-implement window is a rectangular window represented by (3.6).

$$w[n] = \begin{cases} 1, & n = 1, 2, \ldots, N \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

However, the selection of a proper window always involves a trade-off between high **frequency resolution** and low **spectral leakage**. On the one hand, convolution with mainlobe smooths

the estimate over nearby frequencies and the frequency resolution is determined by the width of the mainlobe. On the other hand, the sidelobes cause sidelobe energy to appear in the spectrum, i.e. spectral leakage. (from ELEN90058 *Signal Processing* lecture slides by Erik Weyer)

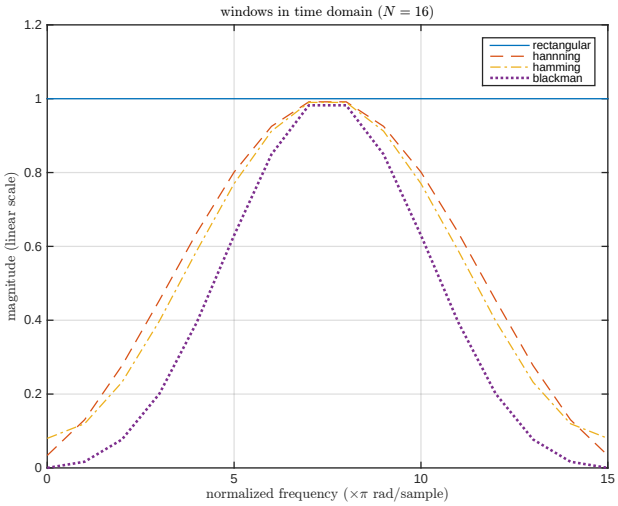


Figure 3.3: Windows in Time Domain

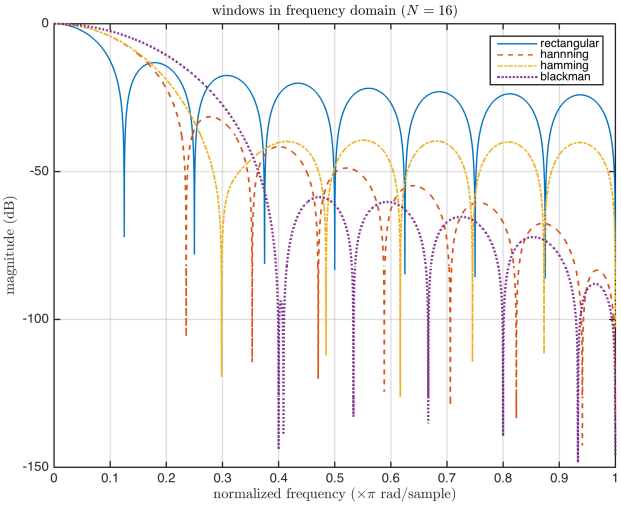


Figure 3.4: Windows in Frequency Domain

Table 3.1: Windows Properties for $N = 16$

| Windows | Rectangular | Hanning | Hamming | Blackman |
|---|---|---|---|---|
| Mainlobe width | $0.125\pi$ | $0.2353\pi$ | $0.2985\pi$ | $0.4\pi$ |
| Peak sidelobe | -13.2 dB | -31.5 dB | -39.8 dB | -58.6 dB |

In terms of the windows involved in Fig. 3.3, Fig. 3.4 and Table 3.1, *rectangular* window has the best frequency resolution (narrowest mainlobe) at the expense of highest spectral leakage (biggest sidelode peak) while *Blackman* window has the lowest spectral leakage (smallest sidelode peak) accompanied by worst frequency resolution (widest mainlobe). Eventually, we choose an intermediate *Hamming* window given in (3.7).

$$w[n] = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right) \quad n = 1, 2, \ldots, N \tag{3.7}$$

where $\alpha = 25/46 \approx 0.54$ and $\beta = 1 - \alpha \approx 0.46$.

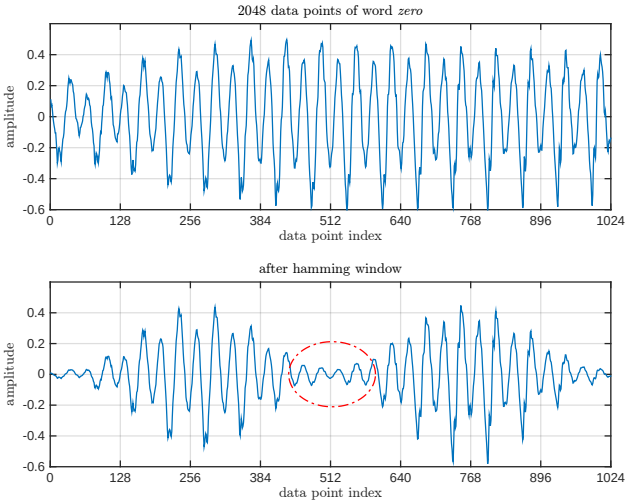


Figure 3.5: Information Loss

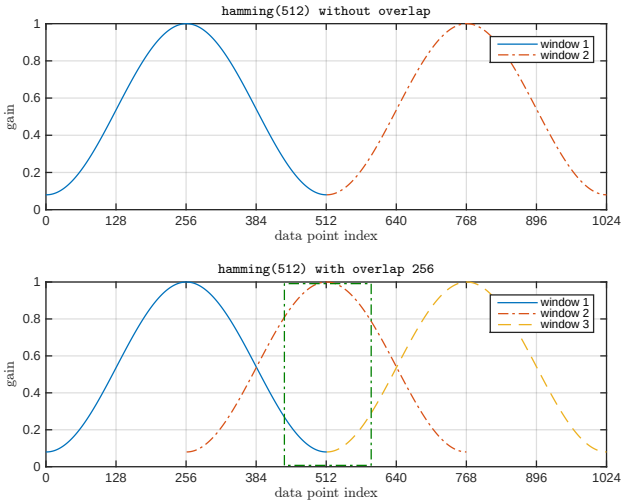


Figure 3.6: Hamming with/without overlap

Fig. 3.5 shows the effect of the framing operation instructed by (3.6) and (3.7) for $N = 512$. The red dash-dot ellipse in subplot 2 demonstrates the loss of information (data points near two borders are severely attenuated) due to the bell shape of the Hamming window shown in Fig. 3.6 subplot 1.

In order to avoid information loss, we overlap each frame by half of the frame size. We choose to overlap a half primarily because the data points most attenuated in current frame will have largest gain in the next frame (shown by the green dash-dot rectangle in Fig. 3.6). Thus, information is effectively preserved.

## 3.4 Thresholds

An important task involved in speech recognition is to distinguish the informative speech (voiced & unvoiced) frames for further processing (MFCC) from the useless silent frames that will be discarded. The basic decision-making strategy is based on the three types of speech signals (voiced, unvoiced & silent) explained in previous section. **Energy** and **zero-crossing count** are two major metrics to determine whether a frame is voiced, unvoiced or silent.

### 3.4.1 Energy

The energy of a finite-length discrete signal $s_j[n]$ $(n = 1, 2, \ldots, N)$ is the sum of the square of the amplitudes. We define the energy of the $j$-th frame

$$E_s[j] = \sum_{n=1}^{N} |s_j[n]|^2 = \sum_{n=1}^{N} (s_j[n])^2 \tag{3.8}$$

### 3.4.2 Zero-crossing count

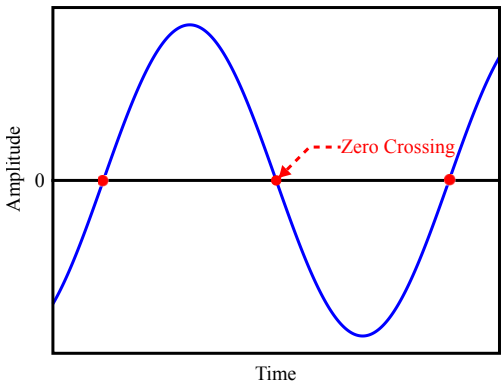


Figure 3.7: Zero Crossing Illustration (from Wikipedia)

As is illustrated in Fig. 3.7, a zero-crossing is a point where the **sign** of a signal changes, represented by a crossing of the time axis where amplitude is zero in the graph. Zero-crossing count is the times of zero-crossing occurrence in a stipulated period. Zero-crossing count of the $j$-th frame can be mathematically represented in (3.9).

$$Z_s[j] = \sum_{n=1}^{N-1} \Big| \operatorname{sgn}(s_j[n]) - \operatorname{sgn}(s_j[n+1]) \Big| \tag{3.9}$$

$$\operatorname{sgn}(s_j[n]) = \begin{cases} 1, & s_j[n] \geq 0 \\ 0, & s_j[n] < 0 \end{cases} \tag{3.10}$$

### 3.4.3 Features of Different Frame Types

We manually identify the unvoiced region and voiced region (the remains are silent region) of several different words by hearing the sound and observing the waveform.

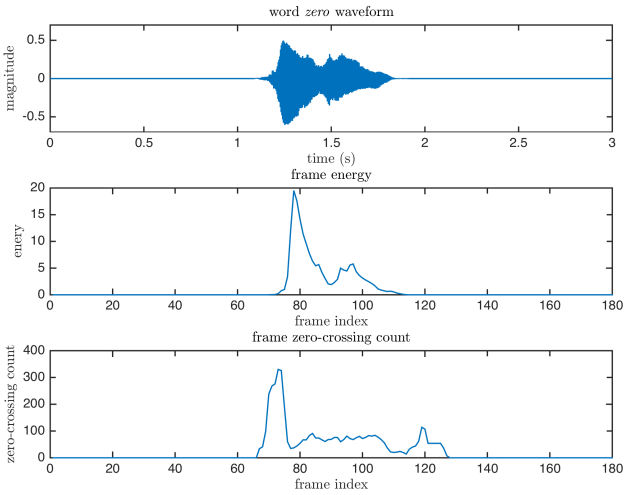


Figure 3.8: Waveform, $E_s$ & $Z_s$

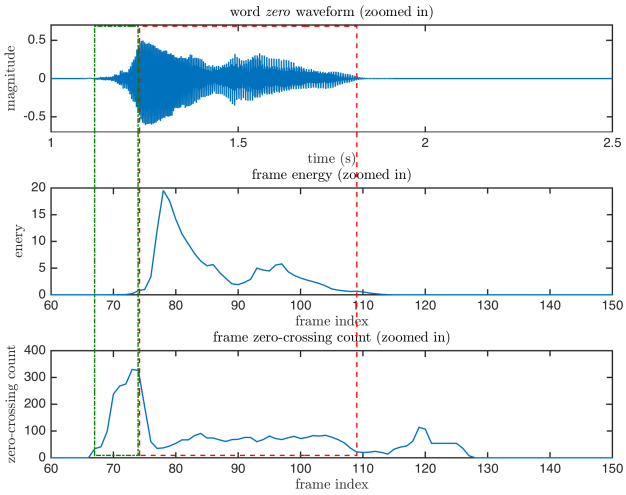


Figure 3.9: Waveform, $E_s$ & $Z_s$ (Zoomed-in)

In the case of word *zero*, Fig. 3.8 shows the waveform and corresponding frame energy as well as the zero-crossing count of each frame. Fig. 3.9 is a zoomed-in version of Fig. 3.8. The green dash-dot rectangle stands for the unvoiced region while the red dashed rectangle indicates voiced region. We can clearly see that unvoiced frames have low energy but high zero-crossing count, voiced frames have high energy but low zero-crossing count while silent frames have not only low energy but also low zero-crossing count. The relationships are summarized in Table 3.2.

Table 3.2: Properties of Different Frame Types

| Type | Energy | Zero-crossing count |
|---|---|---|
| Voiced | high | low |
| Unvoiced | low | high |
| Silent | low | low |

### 3.4.4 Find Thresholds

Fig. 3.10 shows the methods taken to find the energy threshold and the zero-crossing threshold. At the beginning, we manually identify the voiced and unvoiced frames of different words. Then, we calculate the average energy of these voiced frames and take noise level into consideration, finally determine a frame energy threshold. Analogously, we calculate the mean zero-crossing count of unvoiced frames and take the noise frequency into account, eventually

obtain the zero-crossing threshold.

Note that thresholds are subject to the change of recording devices and environment noise. Hence, they have to be calibrated after the system is implemented on the DSP board and integrated with the *Least Mean Square* noise cancellation scheme.

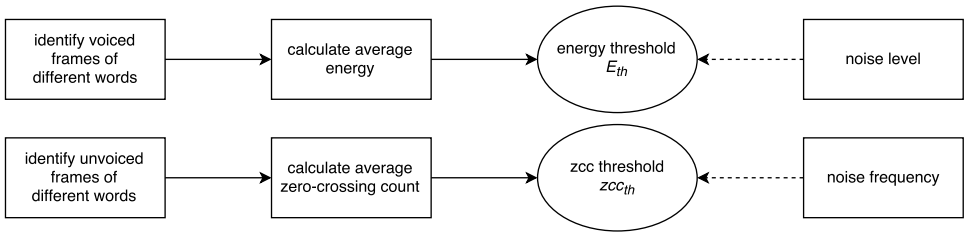


Figure 3.10: Methodology to Find Thresholds

### 3.4.5 Decision Rule

Fig. 3.11 shows the decision-making strategy based on the thresholds obtained from above. Firstly, we calculate the energy of a frame and compare the energy with the threshold. A frame with higher energy is regarded as a voiced frame. Then, compute the zero-crossing count of low energy frame. If zero-crossing count is higher than the threshold, this frame is considered as unvoiced frame; otherwise, this frame will be discarded.

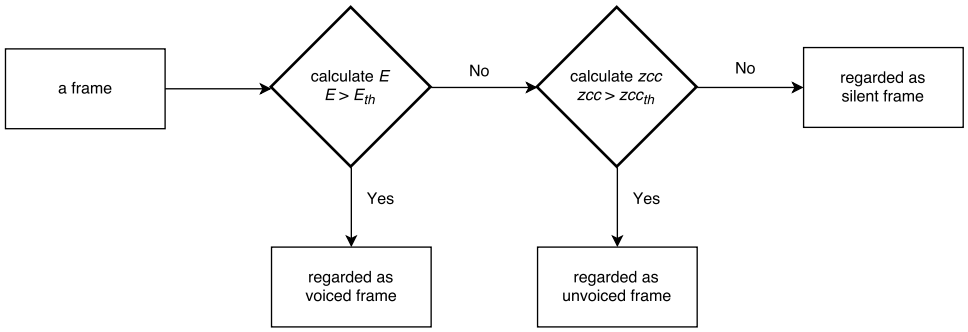


Figure 3.11: Decision Rule

By testing the speech data that we obtained, we discover that 20-50 frames out of 180 will be passed to the next procedure.

## 3.5 Mel-frequency Cepstral Coefficients Extraction

Once informative frames are picked out by discarding the silent frames, the next stage is to calculate the Mel-frequency cepstral coefficients for each frame. MFCC algorithm is based on the concept of power cepstrum. The whole process to obtain power cepstrum defined in (2.4) on page 16 can be divided into three procedures.

1. Compute the Discrete Fourier Transform $S_j[k]$ and corresponding power spectrum $\hat{S}_j[k]$ of a time-domain signal $s_j[n]$.
2. Take the logarithm of the power spectrum $\hat{S}_j[k]$.
3. Conduct inverse Fourier transform.

From previous section, we have known human hearing responds to the entire critical band instead of individual frequencies in this band. Thus, MFCC calculates the total power within each certain mel-scale band prior to log scaling (step 2). Moreover, MFCC substitutes Discrete Cosine Transform for inverse Fourier transform to reduce the computational complexity.

### 3.5.1 Power Spectrum

Discrete Fourier Transform (3.11) constitutes the cornerstone of spectrum analysis.

$$S_j[k] = \sum_{n=1}^{N} s_j[n] W_N^{(n-1)k} \quad k = 1, 2, \ldots, N \tag{3.11}$$

$$W_N = e^{\frac{-2\pi i}{N}} \tag{3.12}$$

It can be clearly seen that the computations required by DFT increase dramatically as length $N$ increases. Due to the high computational requirement, direct implementation of DFT for large sequences has not been feasible. However, Fast Fourier Transform has made implementation of DFT practical in real-time processing.

FFT algorithms commonly implement a divide-and-conquer approach, i.e. an $N$-point DFT is successively divided into smaller DFTs. One of the most popular FFT algorithms is Radix-2 which restricts the sample length to a power of two. Therefore, we chose the frame length $N = 512$ (the 9th power of 2) to make full use of the FFT function. Table 3.3 compares the computational requirements of DFT and Radix-2 based FFT.

Table 3.3: Computational Requirements of DFT and Radix-2 FFT

| | $N$ | $N = 512$ |
|---|---|---|
| DFT multiplications | $N^2$ | 262144 |
| DFT additions | $N^2 - N$ | 261632 |
| FFT multiplications | $\frac{N}{2}\log_2(N)$ | 2304 |
| FFT additions | $N\log_2(N)$ | 4608 |

We have known that the DFT $X[k]$ of a real sequence $x[n] \in \mathbb{R}$ is a conjugate symmetric sequence (from ELEN90058 *Signal Processing* Workshop 3), i.e.

$$X[k] = X^*[\langle -k \rangle_N] = X^*[N-k] \tag{3.13}$$

When computing the power spectrum, we are motivated to use this symmetry property and discard the last $(\frac{N}{2} - 1)$ points. Specifically, $S_j[1]$ is DC component of the signal; $S_j[2]$ to $S_j[\frac{N}{2}]$ are the first half of the complex spectrum; $S_j[\frac{N}{2} + 1]$ is the component at Nyquist frequency (8000 Hz).

$$\hat{S}_j[k] = |S_j[k]|^2 \quad k = 1, 2, \ldots, \frac{N}{2} + 1 \tag{3.14}$$

### 3.5.2 Bank Filtering

The energy within each mel-scale bank can be computed by (3.15).

$$X_j[m] = \sum_{k=1}^{\frac{N}{2}+1} \hat{S}_j[k] H_{mel}[m,k] \quad m = 1, 2, \ldots, M \tag{3.15}$$

where $H_{mel}[m,k]$ is the gain of the $k$-th power spectrum data point within bank $m$.

$$H_{mel}[m,k] = \begin{cases} 0 & f_{d2c}(k) \le f[m-1] \\ \dfrac{k - f[m-1]}{f[m] - f[m-1]} & f[m-1] < f_{d2c}(k) \le f[m] \\ \dfrac{f[m+1] - k}{f[m+1] - f[m]} & f[m] < f_{d2c}(k) \le f[m+1] \\ 0 & f_{d2c}(k) > f[m+1] \end{cases} \tag{3.16}$$

where $f_{d2c}(k) = (k-1) \cdot \frac{F_s}{N}$ transforming the index $k$ of DFT into continuous-time frequency.

$f[m]$ $(m = 0, 1, \ldots, M+1)$ are equally spaced in mel-scale and can be computed by (3.17).

$$f[m] = \text{Mel}^{-1}\left(\text{Mel}(f_{\min}) + m \cdot \frac{\text{Mel}(f_{\max}) - \text{Mel}(f_{\min})}{M+1}\right) \tag{3.17}$$

where the mel-scale to frequency transform $\text{Mel}^{-1}(m_{mel})$ is the inverse function of (2.3).

$$f = \text{Mel}^{-1}(m_{mel}) = 700\left(10^{\left(\frac{m_{mel}}{2595}\right)} - 1\right); \quad f \text{ in Hz} \tag{3.18}$$

We take $f_{\min} = 0$ Hz and $f_{\max} = \frac{F_s}{2} = 8000$ Hz as per the system specification and $M = 20$ according to [15]. $f[m]$ in Hz and corresponding mel-scale are listed in the appendix on page 97.

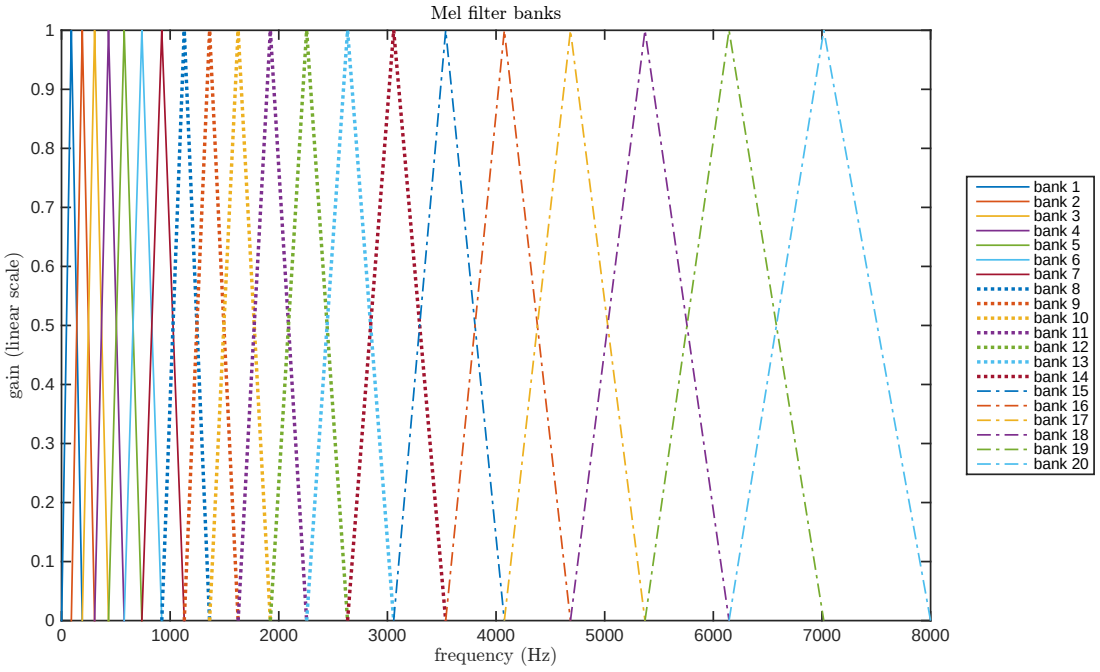


Figure 3.12: Mel Filter Banks

Fig. 3.12 shows that $H_{mel}$ given in (3.16) and (3.17) is essentially a band-pass filter for each bank. $H_{mel}$ offers maximal gain 1 at $f[m]$ (the 'central frequency' of bank $m$ in mel-scale) and then linearly decreases to 0 until reaching adjacent frequency boundaries ($f[m-1]$ and $f[m+1]$) on both sides.

### 3.5.3 Log Scaling

Log scaling is the second step of cepstrum transformation and also makes MFCC algorithm more resilient to both very quiet and very loud sound. Besides, the log scale in dB imitates

human nonlinear perception to loudness [16]. Without taking logarithm, recognition accuracy is severely reduced [17].

$$\hat{X}_j[m] = \log_{10}(X_j[m]) \quad m = 1, 2, \ldots, M \tag{3.19}$$

### 3.5.4 Discrete Cosine Transform

At the last step, IDFT is replaced by a discrete cosine transform (DCT) due to the symmetric and real characteristic of log power spectrum $\hat{X}_j[m]$ [12, 18].

$$\hat{C}_j[n] = \sqrt{\frac{2}{M}} \sum_{m=1}^{M} \hat{X}_j[m] \cos\left(\frac{\pi}{M}(m - 0.5)(n - 1)\right) \quad n = 1, 2, \ldots, F \tag{3.20}$$

The order of DCT ($F$) determines the amount of MFCCs. Higher-order coefficients include excitation information while lower-order coefficients indicate the slowly varying vocal tract. The latter one is more useful for speech recognition.[17] Hence, it is beneficial to condense the $M$-point sequence $\hat{X}_j[m]$ into a shorter $F$-point sequence $\hat{C}_j[n]$ ($F < M$). For example, European Telecommunications Standards Institute adopts $F = 13$ in their speech recognition standard [19]. We test the system performance for $F = 12, 13, \ldots, 16$ and $F = 13$ gives the best recognition accuracy. The choice of $F$ will be further discussed in chapter 6 *Design & Performance*.

## 3.6 Summary

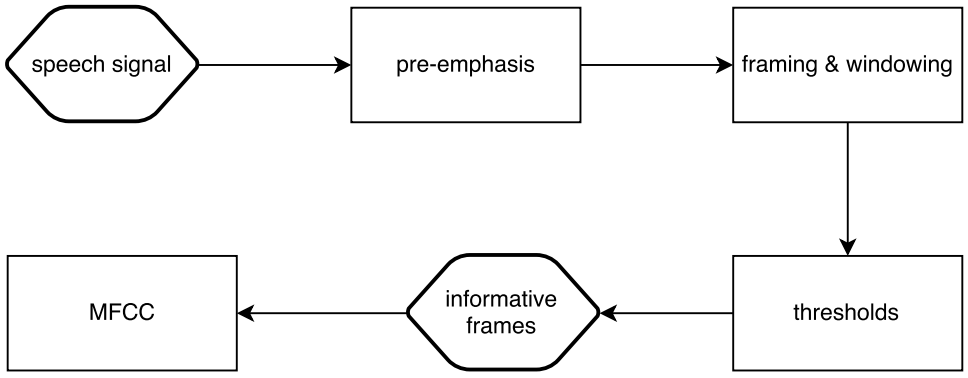


Figure 3.13: Feature Extraction

In order to extract speech features, we firstly pre-emphasize the speech signal. Then, we devide 48000 data points (for 3 seconds) into 180 frames. After the threshold block, we obtain $J$

informative frames out of 180 frames. Sequently, we extract the feature vector $\hat{C}_j \in \mathbb{R}^F$ from each frame. Eventually, a $J \times F$ MFCC matrix is obtained and will be processing by following classfication (Viterbi) block as shown in Fig. 1.1 on page 9.

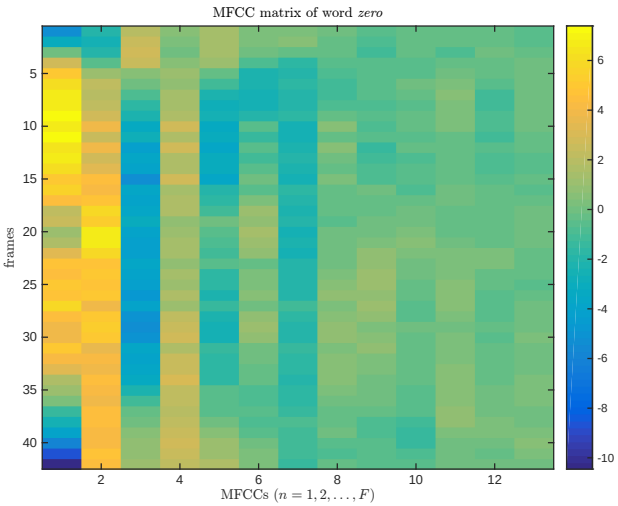


Figure 3.14: MFCC Matrix I of word *zero*

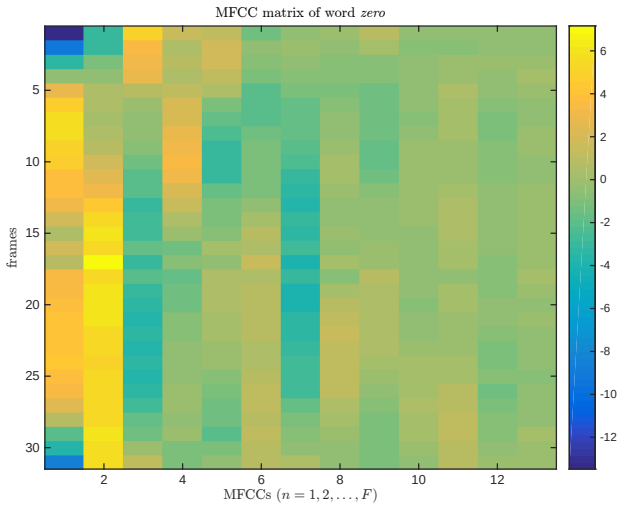


Figure 3.15: MFCC Matrix II of word *zero*

Fig. 3.14 and Fig. 3.15 illustrate the MFCC matrices of word *zero* spoken by two individuals. They have same numbers of columns ($F = 13$), but the numbers of rows are different which depend on how many frames are discarded during the threshold procedure.