SOUND EFFECTOR BASED ON SPEECH RECOGNITION

Yuhui Chen, Shengwen Yang

Department of Electrical and Computer Engineering, University of Rochester

ABSTRACT

Sound effector is used to adjust sounds to better effects. It is widely used in recording studio, music production, and electronic guitar. Traditionally, hardware and software sound effectors are controlled by buttons or sliders. But in this project, we are trying to create a sound effector based on speech recognition so that users could control it by voice. Since the project is based on MATLAB, which is a high-level language, we create a graphical users interface for manipulating. Obviously, it is more convenient to use than before.

Index Terms— sound effector, speech recognition, feature extraction, MFCC, VQ

1. INTRODUCTION

Sound effector is commonly used in electric guitar to implement various sound effects, in recording studio to adjust the timbre of human voice and different musical instruments, also in MP3 player to generate various styles of music, since it makes music pleasant and melodious to hear. Hardware sound effectors usually have several sliders to control the delay, reverb, etc. For software, it often has some preset effects as well, to generate the most commonly used sound effects.

Speech recognition (SR) is the translation of spoken words into text. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT). Nowadays, speech recognition is widely used in vehicle-mounted system, electronic products, such as Siri of Apple, and medical area. It makes our lives easier and more convenient. As the fast improvement of speech recognition, the future of human's life would be smarter and smarter.

Speaker recognition is the identification of the person who is speaking by characteristics of their voices (voice biometrics), also called voice recognition. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on specific person's voices or it can be used to authenticate or verify the identity of a speaker as part of a security process.

In this paper, we try to combine the above three techniques together to create a voice controlling sound effector based on speech recognition.

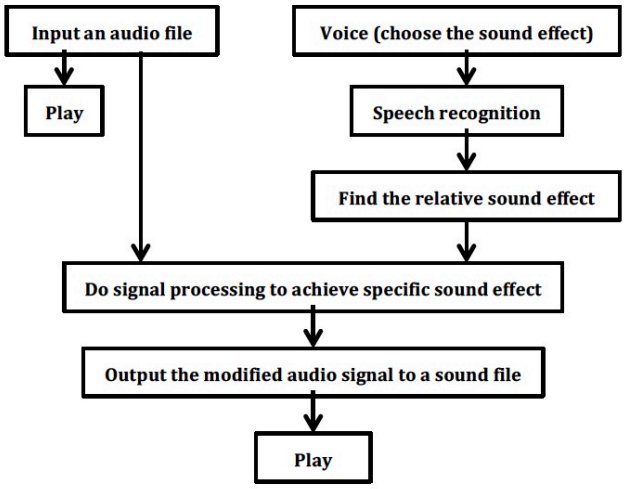

Fig 1. The process of sound effector based on speech recognition

2. SPEECH RECOGNITION

Speech recognition can be further divided into two different tasks - Speaker identification (to determine which one of a group of known voices best matches the input voice sample) and Speaker verification (to determine from a voice sample if a person is whom s/he claims to be). For both of the above tasks, the utterances also have two types - Text dependent (the utterances can only be from a finite set of sentences) and Text independent (the utterances are totally unconstrained).

In generally, speaker identification/verification with text-dependent utterances has a higher recognition rate.

2.1. Feature extraction

In this project, MFCC (Mel-frequency cepstrum coefficients) is used for feature extraction with the following steps:

1. Convert the chosen signal to frames
2. Perform windowing
3. Perform DTFT (Discrete-time Fourier Transform)

And have frequency spectrum:

$$X_a(k) = \sum_{n=1}^{N-1} x(n) e^{-\frac{j2\pi nk}{N}}, 0 \le k \le N-1$$

where $x(n)$ is the input signal, and N is the order.

4. Take square of magnitude of the spectrum, and have energy spectrum.
5. Let energy spectrum passes through a group of triangular filters with size Mel.
6. Define a group of M filters with center frequency f(m),m=1,2,3…M, and M=100.
Logarithm of filtered energy

$$S(m) = \ln\left(\sum_{k=1}^{N-1} |X_a(k)|^2 H_m(k)\right),$$

$$0 \le m \le M-1$$

where $H_m(k)$ is the frequency response of the chosen filter.
7. Calculate MFCC using DCT (Discrete Cosine Transform):

$$C(n) = \sum_{m=0}^{M-1} S(m)\cos\left(\pi n\left(m - \frac{0.5}{m}\right)\right),$$

$$0 \le n \le N-1$$

where MFCC is between 20 and 30, and set the cepstrum coefficient as 20th order.

2.2. Codebook generation
Every speaker is a signal source, referred as a codebook. A codebook includes multiple characteristics of the speaker as long as the sequence is long enough. Here LBG (Linde–Buzo–Gray) algorithm is adopted to create vector quantization (VQ) codebook B, where the sequence is X_k, with the following steps:
1. Suppose B1 is the mean value of the whole feature vectors extracted from all frames.
2. Divide B_m into 2m pieces according to the following equation:

$$B_m^+ = B_m(1+\mathcal{E})$$

$$B_m^- = B_m(1-\mathcal{E})$$

where $\mathcal{E} = 0.01$, m starts from 1.
3. Classify the sequences obtained before, and calculate the summation of vector quantization errors (distortion), which is D, and the relative errors according to the following formula:

$$D^{(n)} = \sum_{k=1}^{K} mind(X_k, B)$$

$$\left|\frac{D^{(n-1)} - D^n}{D^n}\right|$$

4. Calculate the new mean value of the whole feature vectors extracted from all frames, and repeat step 3.
5. Repeat step 2, 3, 4 until having the codebook of M codes, where $D_0 = 10000$.

2.3. Speaker recognition
Suppose the feature vector of an unknown speaker is $\{X_1, K, X_t\}$, where there are T frames belong to codebook. Calculate the mean value quantization errors D, and set a threshold. If D< threshold, the speaker verified, otherwise, it's not the target speaker.

$$D = \frac{1}{T\sum_{j=1}\min\left[d\left(x_j, B_m\right)\right]}, 1 \le m \le M$$

3. SOUND EFFECTS

3.1 Echo
Echo effect is fundamentally delaying. Delay is an audio effect which records an input signal to an audio storage medium, and plays it back after a period of time. The delayed signal may either be played back multiple times, or played back into the recording again, to create the sound of a repeating, decaying echo. In MATLAB, the audio signal is saved as an array, so we could get the output signal from the following equation:

$$y[n] = x[n] + x[n-m]$$

where m is the delay parameter. In this system, we set m as 0.1 seconds.

3.2 Bass
The basic idea to get bass effect is to increase the intensity of the low frequency. To separate the low frequency from the whole frequency range, we apply the low pass filter with 1000 Hz pass-band stop frequency to the input audio signal. Then increase this low frequency part to two times, and add to the original signal to get the bass effect. The output is:

$$y[n] = x[n] + 2x[n] \cdot f[m]$$

where f is a low-pass filter.

3.3 Mix
Using mix effect, we could add background sound to the original audio. The formula is as follows:

$$y[n] = x[n] + b[n]$$

where b is the background sound.
There is a default bird audio file used in this system. Since it is only one-second long, one way to extend it to the same length as the original audio file is to add it circularly.
Meanwhile, the intensity of the two sound files also need to be adjusted carefully to avoid the mixture of the background and foreground sound.

3.4 Pitch change

Pitch change is fundamentally resampling, where sample-rate conversion is the process of changing the sampling rate of a discrete signal to obtain a new discrete representation of the underlying continuous signal. Resampling includes down-sampling and up-sampling. Down-sampling is to make a digital audio signal smaller by lowering its sampling rate. Up-sampling is the opposite. The default value resampling is down-sampling in this system. The process is shown below:

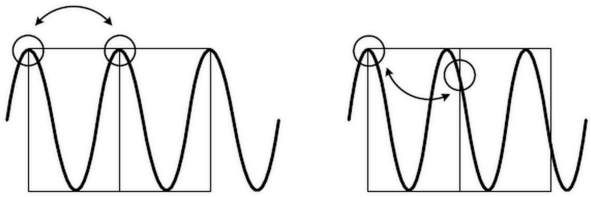

Fig 2. The principle of resampling

3.5 Panning

Panning is the distribution of a sound signal (either monaural or stereophonic pairs) into a new stereo or multi-channel sound field determined by a pan control setting. The left and right channels formula for amplitude panning is:

$$y[left] = x \cdot \cos\left(\lambda \frac{\pi}{2}\right)$$

$$y[right] = x \cdot \sin\left(\lambda \frac{\pi}{2}\right)$$

By adjusting λ from 0 to 1, we could get the panning effect that the sound gradually from left channel to right channel.

4. IMPLEMENTATION

In order to manipulate the sound effector created, a Graphical Users Interface (GUI) is generated as:

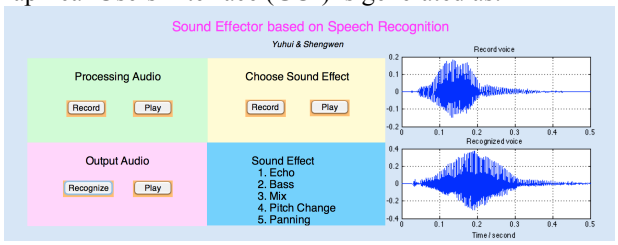

Fig 3. Graphical Users Interface for manipulating

It works according to the following steps:

1. Press ‘Record’ button to record a 10-second piece of sound as the original processing audio or just use the default one, and ‘Play’ button is used to play the processing audio file.
2. Record a 0.5-second long audio signal to choose a sound effect listed below: echo, bass, mix, pitch change, or panning. There will be a message box that shows ‘Finish recording’ if you have run out of time. If click ‘Play’ button in this part, the record sound can be heard while its waveform reveals on the right as ‘Record voice’. By checking the waveform, we could find whether the record is successful.
3. Press ‘Recognize’ button to implement speech recognition. The record audio would be compared with all the audio files in the test library, and find which effect is matched. It takes more than three seconds. The recognize result pops up in a message box and the matched audio file in the library would also be plotted on the right as ‘Recognized voice’. The system would apply the chosen sound effect to the processing audio, outputting and saving to the ‘Output.wav’. Click the ‘Play’ button, the final output sound effect will be heard.

5. ERROR MEASUEMENT

5.1 Recognition error

The speech recognition would get the MFCC of the input audio and the audio files in test library and compare it. To visualize the recognize process, we record a matched audio with ‘1.wav’ in the test library and a unmatched audio and compare both of their MFCCs with the MFCC of ‘1.wav’. The comparison of blue line referred as the same MFCC of ‘1.wav’ while the red line referred as the MFCC of the matched and unmatched record audio, and the green represents the distance of input audio and the saved one, is shown below:

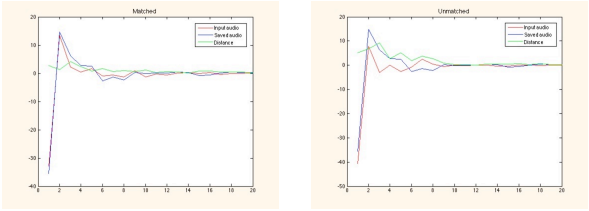

Fig 4. MFCC comparison

According to the figure above, the distance of the unmatched one is significantly larger than that from the matched pair. By measuring the shortest distance between the input audio and the audio in the library, speech recognition has been completed.

5.2 System error rate

We test the error rate of the system by measuring a pair of input voices from both the current person and the saved one.

Table System error rate test

	Right	Wrong	Total
Same person	10	0	10
Different person	0	10	10

From the table, it can be summarized that the generated system is able to successfully recognize all of the voices of

the person with the same features saved in the test library. Otherwise, it fails.

6. CONCLUSION

Sound effector with effects of Echo, Bass, Mix, Pitch change and Panning performs well in programming with each mature and modified algorithm, which contributes to significant effects on default audio file.

In speech recognition section, MFCC and VQ techniques have been frequently used. These two techniques guarantee the accuracy of feature extraction and speaker recognition. The coding of all the techniques mentioned above has been programmed with MATLAB. Results of error measurement reveals that the combination of MFCC and VQ algorithm gives the best performance and also accurate results in most of the cases with an overall efficiency of 90%.

Future research could be focused on a larger database saving multiple vocal orders generated by people with various speech features, which may make it a more powerful voice control system.

7. REFERENCE

[1] John Wiley & Sons. (2011) DAFX - Digital Audio Effects (Second Edition)

[2] Speech recognition, Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc., 14 Mar. 2015. Web. http://en.wikipedia.org/wiki/Speech_recognition

[3] Effects unit, Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc., 28 Feb. 2015. Web. http://en.wikipedia.org/wiki/Effects_unit

[4] Developing an Isolated Word Recognition System in MATLAB, Daryl Ning, http://www.mathworks.com/company/newsletters/articles/developing-an-isolated-word-recognition-system-in-matlab.html