Audio Restoration Based on DSP Tools

EECS 451 Final Project Report

Nan Wu
School of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI, United States
wunan@umich.edu

Abstract**—this report shows the author's work on audio restoration based on DSP tools from and outside of the course of EECS 451, such as moving average system, spectral subtraction method, and LMS adaptive filter design, in order to repair the degraded sound signals by reducing impulse interference, echo effects and background noise. These approaches are achieved on MATLAB, and have shown impressive results on repairing imperfect real life audio recordings.**

Keywords—audio restoration; DSP; LMS; spectral subtraction

I. Introduction

Audio restoration is a generalized term to describe the process of removing imperfections in sound recordings, such as hisses, impulse interference, background noise, echo effects, etc. There are many reasons that imperfections occur. Recordings are possibly damaged due to poor hardware, noisy environment, intended interfering, and improper storage. Among different kinds of imperfections, impulse interference, background noise, and echo effects are pretty common.

Digital Signal Processing (DSP) techniques are usually applied for modern audio restoration. For example, DFT is used to better analyze the characteristics of the signals in frequency domain; filter design is for processing the damaged speech/music signal with some kind of system to get rid of the noise or enhance the desired part. In this project, the moving average system, high pass filter, LMS adaptive filter design and the spectral subtraction method are introduced to deal with specific frequency interference, echo effects and wide band background noise.

II. Impulse Interference Cancelling

Impulse interference is a kind of noise (sounds like clicks and pops) that occurs at specific frequency points, possibly caused by electromagnetic interference and scratches on recording disks. Some of them are periodic, and some are not. Strong impulse may impair the quality of sounds.

To enhance a signal with impulse interference, an impulse noise filter is commonly used. In image processing theories, median filter is a fundamental way to average out the impulse pixels; however we cannot directly use this method for audio signals, because our ears are more sensitive to high frequency signals than eyes. Hence other approaches are needed.

In Figure 1, we can see that the impulses along the time axis repeat irregularly; from the spectrogram it can be concluded that periodic components exist for impulses. What's more, most energy concentrates on low and medium frequencies.

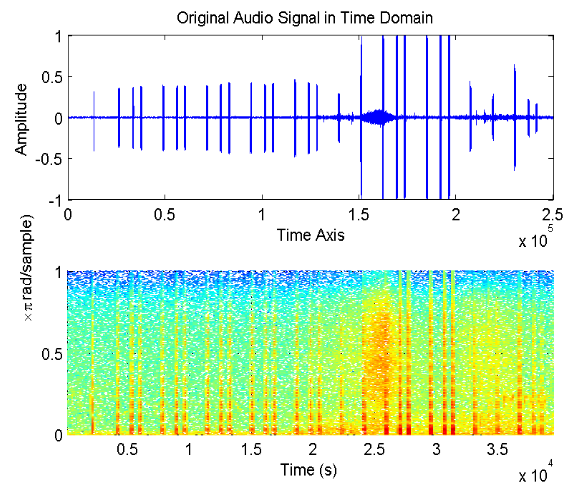

Fig. 1. Original signal representation

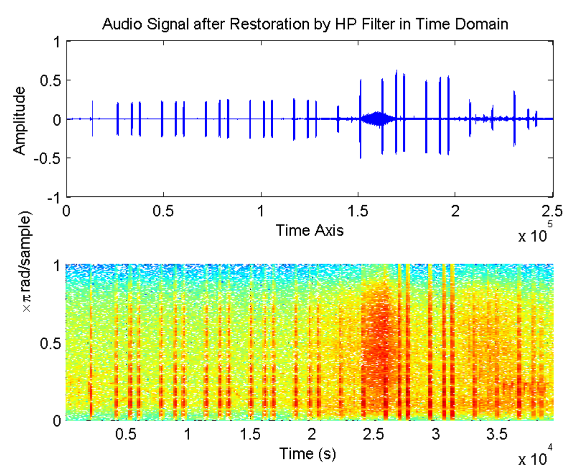

Fig. 2. Original signal after highpass filter ($\omega_p = 0.1\pi$)

A. High pass filter design

The first attempt is the digital filter design. By Analyzing the corrupted signal, it is concluded that most energy of the impulses are located in lower frequencies. Thus we plan to use a high pass filter to suppress those peaks we don't want.

Based on experiments the pass band start point is settled at $\omega_p = 0.1\pi$. The magnitude response of the high pass filter is shown below in Figure 3.

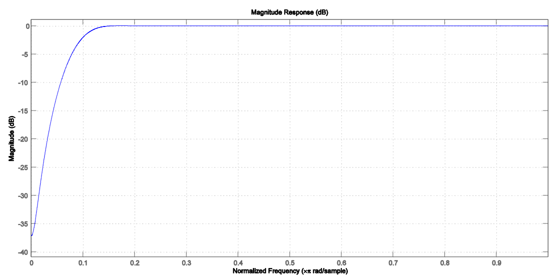

Fig. 3. Magnitude response of the highpass filter

The result of implementing high pass filter is shown in Figure 2. It can be seen that the impulses are attenuated but still exist, and we can still hear the impulse interference after the restoration. Hence this method does not work well.

B. Smoothing system

The second attempt is to revise the median filter for image restoration, combined with the moving average system we learned in class. In this method, we first try to find the positions when impulses occur, by testing the amplitude increments. When the difference between two neighboring points is larger than the threshold, the posterior value is replaced by the average of fixed length prior values. The rest values of the signal remain the same. The smoothing system can be illustrated as

$$y(i) = [\, y(i-1) + y(i-2) + \ldots + y(i-K) \,] / K \qquad (1)$$

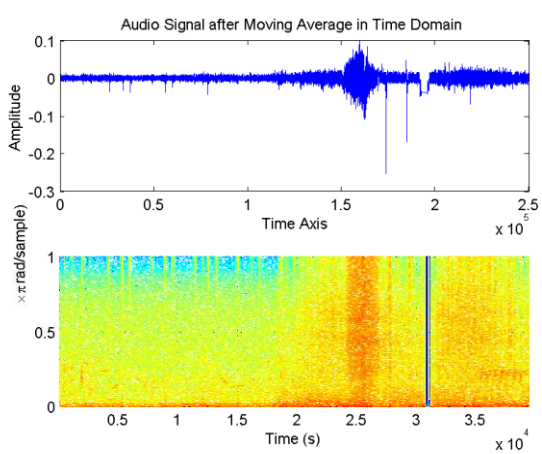

Fig. 4. Signal after highpass filter

Figure 4 shows the result of the signal restored by the smoothing system. In my simulation, we fix the threshold as 0.02, and K equals to 10.

Comparing with Figure 2, it can be concluded that the music part is well protected and many harmonic interference pulses have been reduced or hidden. In addition, the hearing effect is much better than the previous one, with only one small silence duration, which is represented as a blue stripe in the spectrogram.

III. Background Noise Cancelling

Background noise is common in real world. Examples of background noise are environmental noises such as waves, traffic noise, and people talking. A weak background noise is okay for hearing effects, and the existence of background noise would possibly make people calm and comfortable. However, if it is strong enough to cover the music/speech signal, we need to attenuate it.

It is much different to process background noise from narrow band interference and impulse interference, mostly with lower power, zero mean, and a small auto-correlation coefficient. Background noise often has a wide bandwidth along the frequency axis, which often overlaps with the band of speech/music signal. Thus normally we cannot 'filter' it, but spectral subtraction is a useful technique for cancelling background noise.

A. Spectral subtraction method

The concept of the spectral subtraction method is based on the estimation of noise. Peoples' ears are not sensitive to the phases of signals, so that we can estimate the noise signal from a time segment due to its property of short time stationary. Then subtract the estimation from the mixed signal, we can get an approximate clear speech/music result.

The algorithm is illustrated as follows:

- Calculate the noise spectrum from data where only noise exists. Because for short time duration, the signal segment can be considered stationary and we can estimate the whole noise from this pure noise sample.
- Break the signal into several frames (with the same length as the noise segment), and each frame is windowed by hamming window, in order to minimize leak of spectrum after FFT.
- Compute FFT of each windowed frame, and compare its absolute magnitude with that of noise. If it is larger, keep $S(k) - N(k)$ as the estimation of $X(k)$, otherwise the value of $X(k)$ is zero. The phase of $X(k)$ equals to the phase of $N(k)$.
- Compute IFFT of $X(k)$, which is the estimation of clear signal.

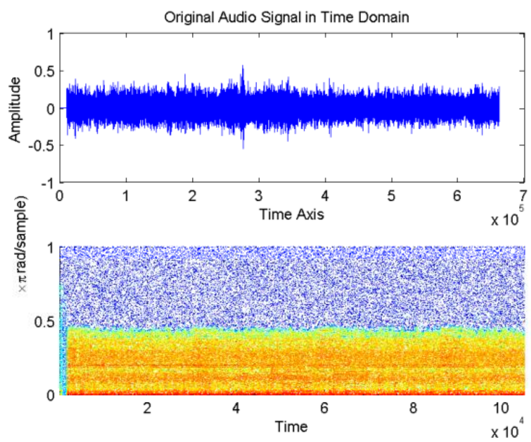

Fig. 5. Original signal before restoration

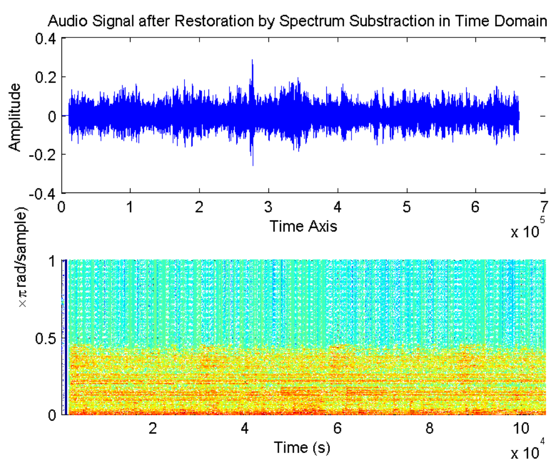

Fig. 6. Signal after spectral subtraction

Achieving the algorithm on MATLAB, the plot of original audio signal and the restored signal is shown in Figure 5 and Figure 6.

From the plot we can see that the harmonic notes are clearer. According to the sounds played out, we can tell that the background noise has been greatly reduced. However, a distinct fresh noise happens. This new noise is caused by the algorithm itself, called musical noise.

B. Improvement of spectral subtraction method

A slight improvement can be used to the algorithm, that is, instead of calculating the absolute value difference between the frame and the noise, compute the difference between signal powered with *a*, and a *b* times noise *N(k)* with power *a*. The improved algorithm can be depicted as the equation

$$S\_new(k)=[S(k)^a-b*N(k)^a]^{(1/a)} \tag{2}$$

Trying to determine the best *a* and *b* (for the audio file used, *a=0.4* and *b=0.9*), we can slightly reduce the musical noise. The signal after improved spectral subtraction method is shown in Figure 7.

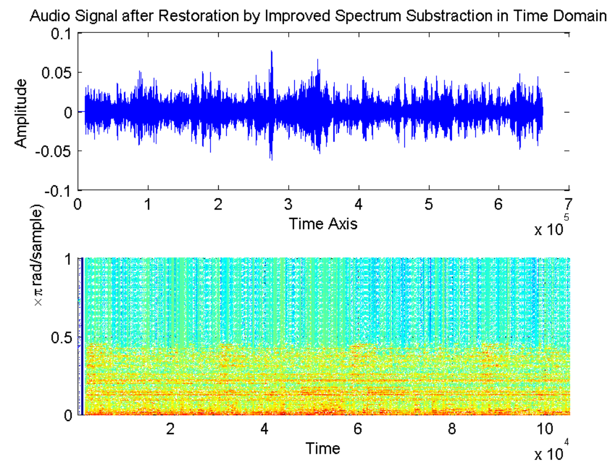

Fig. 7. Signal after improved spectral subtraction

This improved method works really well on signals with additive Gaussian white noise, which is a typical model in simulations (The result is not included in this report but in the attached documents.). However, real life is much more complicated. To better cancel wide band background noise, we need to find more effective approaches to deal with the vestigial 'music noise'.

IV. ECHO EFFECT CANCELLING

Echo effects, also known as delay effects, are normally produced on purpose to make some interesting sound effects, especially common in music products, movie making, etc. The theory is simple to understand, that records play a signal back after a period for multiple times, to create the sound of a repeating decaying echo.

In terms of audio restoration, sometimes the echo effects may impair the hearing effect of audio files. People may not distinguish the information. Thus to de-echo the speech/music signals is required.

A. LMS adaptive filter design

Least mean square (LMS) is a classic algorithm to mimic a desired filter by adaptively finding the coefficients of the filter in order to make sure the error, which is estimated as the mean square, is minimized. The algorithm is similar as Wiener filter design, with the most important difference that replacing the expectation of estimation with instantaneous estimation. This idea results in the stochastic gradient descent.

The algorithm is illustrated as follows:

- A parameter μ, called the step coefficient, is initialized before the algorithm starts. The value of μ will influence the robustness of the algorithm.

- Initialize the filter coefficient vector $\boldsymbol{w}$, calculate the error vector $e(n) = d(n) - \boldsymbol{w}^T_{2n}\boldsymbol{x}_{2n}$, where $\boldsymbol{d}$ is the desired signal, and x is the input signal. For the goal to reduce the echo effects, we can make $\boldsymbol{d} = \boldsymbol{x}$.

- Update the filter coefficient vector $\boldsymbol{w}$, $w(n+1) = w(n) + \mu e^T x$. Then repeat the second step.

B. Simulation on MATLAB

Dealing with an echoed signal using LMS adaptive filter, the result is shown below in Figure 8 and 9.

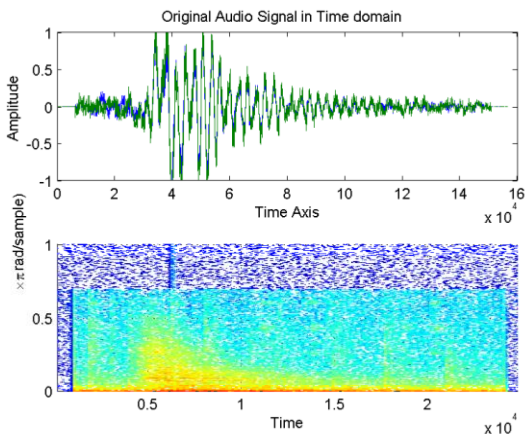

Fig. 8. Original signal representation

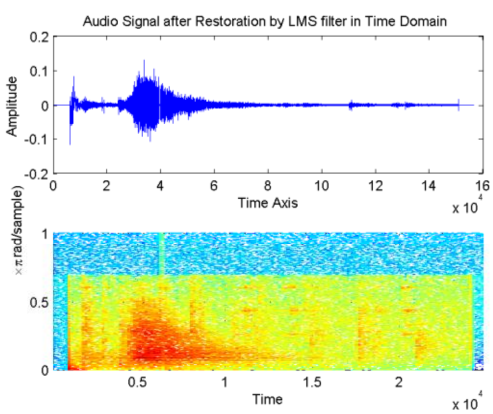

Fig. 9. Signal after LMS adaptive filter

We can tell that after the adaptive LMS filter the time delays of the signal have been cancelled, which are showed as a less loose version of time domain representation. In spectrogram, there are more high frequency components.

Comparing the two sounds before and after restoration, the second is much more clean than the first one.

It is also possible to show the magnitude response of the final LMS filter (Shown in Figure 10).

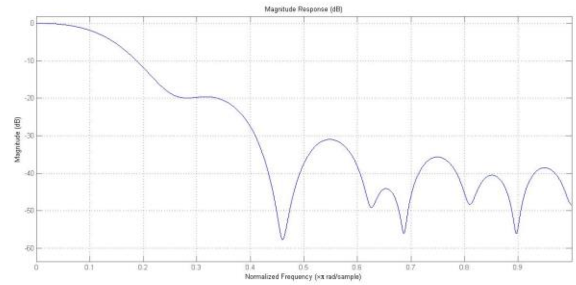

Fig. 10. Magnitude response of the LMS filter (μ=0.009)

V. Conclusions

In this project, we come up with some techniques to enhance the quality of audio corrupted with impulse interference, background noise and echo effects. Based on simulations on MATLAB, impressive results are present in this paper.

For those methods we utilized to restore the damaged signals, some are strong enough but some are limited.

- Smoothing system works better than high pass filter, and most interference can be hidden to people's ears. However, due to the low SNR of some duration, silence segments are made due to the processing. Further work should be focus on solving this problem.

- Noise spectral subtraction method is a fundamental technique to reduce background noise, but the musical noise is caused, which is beyond our expectation. Possibly the algorithm needs further revise to get rid of this phenomenon.

- LMS adaptive filter works well for de-echo a signal. Actually according to my research on this topic, LMS has more implementation future. It has potential dealing with some unknown effect or noise of the audio signals.

During the semester working on this project, I have a great experience dealing with signals in time domain and frequency domain, with all kinds of DSP tools and MATLAB commands, such as FFT, filter design command, and figure command. I become familiar with programming on MATLAB and get a deeper knowledge of the EECS 451 course because of the project, and has gained a great enthusiasm further studying on signal processing.

Acknowledgment

The author thanks Professor Laura Balzano for carefully explaining on all the knowledge in EECS 451 and advice on the ideas of the project, classmate Kaichang Wang, for help working on programming problems.

References

[1] Alan V. Oppenheim, Ronald W. Schafer Discrete-Time Signal Processing (3rd Edition) 2009

[2] Haykin, Simon S. Adaptive filter theory. Pearson Education India, 2008.

[3] Godsill, Simon, Peter Rayner, and Olivier Cappé. Digital audio restoration. Springer US, 2002.

[4] Wikipedia, http://en.wikipedia.org/wiki/Audio_restoration

[5] Wikipedia, http://en.wikipedia.org/wiki/Least_mean_squares_filter

[6] Freesound.org, https://www.freesound.org/

[7] Audio Archive, https://archive.org/details/audio.

[8] Boll, Steven. "Suppression of acoustic noise in speech using spectral subtraction." Acoustics, Speech and Signal Processing, IEEE Transactions on27.2 (1979): 113-120.

[9] http://www.lms.lnt.de/en/research/activity/audio/signal/nlaec.php