Dereverberation on Music Signals

Yinan Zhou
yinan.zhou@mail.mcgill.ca

Abstract. This project implements a block-based dereverberation algorithm. The hyperparameter settings are explored on a singing voice recording. From the results, it can be concluded that the amount of dereverberation is proportional to the values of B, α, $MaxValue$ and $Bias$. In contrast, it is inversely proportional to the values of γ and $MinGain$. To get a natural result, the settings of γ and α are mainly important. Moreover, the algorithm is also implemented on a short clip of a symphonic music piece by Mozart. It is found that this algorithm has limited effect on signals with high reverberation. The hyperparameter setting for high reverberation extraction is also discussed.

1 Introduction

In general, audio signals have two components: the direct component and the reverberant component. The direct component is the original dry signal produced by the sound source. The reverberation component is the result of the original dry signal passing through the reverberation system of the given space. It is worth noting that the reverberate component is dependent on the original dry signal. Moreover, the information of the reverberation system is usually not accessible. The recordings capture the direct sound together with the acoustic characteristics of the space. Therefore, it is extremely difficult to recover the direct component once the reverberant component has been superimposed on it.

Dereverberation can be applied to many situations in real life. For example, the reverberation of a conference room may make the speech signal undesirably hollow during a teleconference. Besides, speech signals for security and surveillance can be highly unclear because of reverberation. Dereverberation algorithms can increase the clarity of the sound in similar situations.

This project implements the dereverberation algorithm proposed by Gilbert A. Soulodre in 2010 [1]. This method subtracts the estimation of the magnitude of the reverberant component from the input signal. The phase of the dry component is approximated by the phase of the input signal. To make the process inaudible, the algorithm is performed in a block-based manner. In addition, the effects of different hyperparameter settings on monophonic music signals are explored. Furthermore, the algorithm is implemented on a short clip of symphonic music, respectively with average and high reverberation.

2 Dereverberation

2.1 Primary Assumptions

The dereverberation algorithm relies on three primary assumptions. First, over short time intervals, human auditory systems are relatively insensible to phase differences or errors. Second, in a reverberation system, nearby impulse responses have similar magnitude responses but very different phase responses. Hence, if the process is done in a short period of time, the phase of the input signal can be regarded as an acceptable approximation for the phase of the original dry signal and for the phase of the reverberant signal.

2.2 Block-Based Structure

With the primary assumptions, the dereverberation process should only change the magnitude of the input signal $m(t)$ in a block-based manner. To alter the magnitude, the idea is to determine a gain vector for the frequency response $M(\omega)$. For a reverberation system, the impulse response $h(t)$ can be approximated by a Finite Impulse Response (FIR) filter of sufficient length. The input signal $m(t)$ can thus be obtained by convolving the dry component $s(t)$ with $h(t)$:

$$m(t) = s(t) * h(t). \tag{1}$$

Then, the impulse response $h(t)$ can be divided into $B+1$ blocks: $h_0(t), h_1(t)$, ..., $h_B(t)$. The corresponding frequency domain representations are $H_0(\omega), H_1(\omega)$, ..., $H_B(\omega)$. All the blocks are of the same length D. Therefore, block-based FIR filter process can be represented by:

$$m(t) = s(t) * h_0(t) + \sum_{i=1}^{B} s(t) * \delta(t - iD) * h_i(t), \tag{2}$$

where $s(t) * h_0(t)$ includes the direct signal component. The second part is the reverberant component $r(t)$ of the input signal. In the direct component, the

reverberant portion $h_0(t)$ will be inaudible to human beings if D is short enough. Therefore, the direct component can be considered as an estimation of the dry signal. In the frequency domain, the process can be represented as:

$$\begin{aligned} M(\omega) &= S(\omega)H_0(\omega) + \sum_{i=1}^{B} S(\omega)z^{iD}H_i(\omega), \\ &= S(\omega)H_0(\omega) + R(\omega), \end{aligned} \tag{3}$$

where the first half is the direct component, and the second half is the reverberant component $R(\omega)$.

To undo the FIR filter effects, an Infinite Impulse Response (IIR) filter can be applied. In real-life situations, however, it is extremely difficult to measure the exact impulse response of a reverberation system. Therefore, the perceptually relevant estimates $\tilde{H}_0(\omega), \tilde{H}_1(\omega), ..., \tilde{H}_B(\omega)$ can be used to derive an estimate of $S(\omega)$. These estimates only operate on the magnitudes, which can be taken as:

$$\tilde{H}_i(\omega) \approx |H_i(\omega)|^2. \tag{4}$$

2.3 Impulse Response Estimation

To extract the dry signal, the impulse response of the reverberation system first need to be estimated. As discussed above, the reverberant effect in the direct component is negligible. That is:

$$\left|\tilde{H}_0(\omega)\right|^2 = 1. \tag{5}$$

Both decay rates of the dry signal $S(\omega)$ and the reverberation system determine the decay rate of the input signal $M(\omega)$ at a certain frequency. The decay rate of the reverberation system at a given frequency is fairly stable over time. The decay rate of the dry signal, however, varies continuously. Therefore, the decay in the input signal $M(\omega)$ is mostly dependent on the reverberation system decay. Moreover, when the dry signal stops at a certain frequency, the input signal decays at the fastest rate. This is considered as the best opportunity to estimate $\left|\tilde{H}_i(\omega)\right|^2$. At this point, only the reverberant component comes next. The reverberation system decay can thus be obtained. Therefore, $\left|\tilde{H}_i(\omega)\right|^2$ can be estimated as:

$$|C_i(\omega)|^2 = \begin{cases} \frac{|M_0(\omega)|^2}{|M_i(\omega)|^2}, & \frac{|M_0(\omega)|^2}{|M_i(\omega)|^2} < \left|\tilde{H}_i(\omega)\right|^2 \\ \left|\tilde{H}_i(\omega)\right|^2 Bias(\omega) + \epsilon, & otherwise \end{cases} \tag{6}$$

with $i = 0, 1, ..., B$. In this equation, $Bias(\omega)$ is a value larger than 1.0. It prevents $|C_i(\omega)|^2$ from being stuck at an inappropriate minimum value. ϵ is a small positive value that prevents the situation of zero.

To get a natural result, it is practical to limit $|C_i(\omega)|^2$:

$$|C_i(\omega)|^2 = \begin{cases} MaxValue_i(\omega), & |C_i(\omega)|^2 > MaxValue_i(\omega) \\ |C_i(\omega)|^2, & otherwise \end{cases} \tag{7}$$

with $i = 0, 1, ..., B$. $MaxValue_i(\omega)$ limites the maximum value of $|C_i(\omega)|^2$ and prevents artifacts from appearing in the extracted dry signal.

A temporal smoothing can also be applied to obtain a more stable estimation:

$$|\tilde{H}_{i,\tau}(\omega)|^2 = \alpha_i(\omega)|\tilde{H}_{i,\tau-1}(\omega)|^2 + (1 - \alpha_i(\omega))|C_i(\omega)|^2, \tag{8}$$

where τ is the current time frame and $\alpha_i(\omega)$ controls the amount of smoothing. The higher the value of $\alpha_i(\omega)$ is, the larger the amount of temporal smoothing is.

2.4 Gain Vector

As mentioned above, an IIR filter can be applied to undo the reverb effect. The structure is based on the relevant estimation of the impulse response $|\tilde{H}_i(\omega)|^2$. The process for a given block can thus be represented as:

$$\tilde{S}_0(\omega) = \frac{M_0(\omega) - \sum_{i=1}^{B} \tilde{S}_i(\omega)\tilde{H}_i(\omega)}{\tilde{H}_0(\omega)}, \tag{9}$$

where $S_0(\omega)$ indicates the dry component in the current block, while $S_i(\omega)$ represents it in the previous ith block. In addition, the reverberant effect in the direct component can be neglected. Therefore, Eq.(9) can be described as:

$$\begin{aligned} \tilde{S}_0(\omega) &= M_0(\omega) - \sum_{i=1}^{B} \tilde{S}_i(\omega)\tilde{H}_i(\omega), \\ &= M_0(\omega) - \tilde{R}_0(\omega), \end{aligned} \tag{10}$$

In this way, the dry component in the current block is obtained. As mentioned in the primary assumptions, the reverb extract process should only change the magnitude of the input signal, thus giving:

$$|\tilde{S}_0(\omega)|^2 = \left| M_0(\omega) - \sum_{i=1}^{B} \tilde{S}_i(\omega)\tilde{H}_i(\omega) \right|^2, \tag{11}$$

which can be approximated by:

$$|\tilde{S}_0(\omega)|^2 = |M_0(\omega)|^2 - \left| \sum_{i=1}^{B} \tilde{S}_i(\omega)\tilde{H}_i(\omega) \right|^2. \tag{12}$$

Hence, the gain vectors for the dry component $G_S(\omega)$ can be calculated by dividing the dry signal estimation with the input signal:

$$\begin{aligned} G_S(\omega) &= \frac{\left|\tilde{S}_0(\omega)\right|^2}{\left|M_0(\omega)\right|^2}, \\ &= 1.0 - \frac{\sum_{i=1}^{B} \left|\tilde{S}_i(\omega)\right|^2 \left|\tilde{H}_i(\omega)\right|^2}{\left|M_0(\omega)\right|^2}. \end{aligned} \tag{13}$$

To make $G_S(\omega)$ always positive, a limitation on the minimum value is applied to it:

$$G_S(\omega) = \begin{cases} MinGain(\omega), & G_S(\omega) < MinGain(\omega) \\ G_S(\omega), & otherwise \end{cases} \tag{14}$$

Furthermore, a temporal smoothing can also be conducted to refine $G_S(\omega)$:

$$G'_{S,\tau}(\omega) = (1 - \gamma(\omega)) * G'_{S,\tau-1}(\omega) + \gamma(\omega) * G_{S,\tau}(\omega), \tag{15}$$

where $\gamma(\omega)$ controls the amount of smoothing and τ indicates the current time frame. With the gain factor, the estimated dry component can be described as:

$$\tilde{S}_0(\omega) = G'_{S,\tau}(\omega) M_0(\omega) \tag{16}$$

Then, $\tilde{S}_0(\omega)$ is converted to the time domain. This process repeats for each time frame to yield the dry signal estimation $s(t)$.

3 Implementation on Music Signals

3.1 Music Signal Source

In this project, the dereverberation algorithm is implemented on music signals. The hyperparameter setting exploration uses the IR Data and the Anechoic Data in the Open Acoustic Impulse Response (Open AIR) Library by University of York [2]. The IR Data contains acoustic impulse responses from buildings, spaces, and other sources. The impulse response used in hyperparameter setting exploration is the impulse responses of Genesis 6 Studio. The Anechoic Data contains sounds recorded in an anechoic environment. The anechoic singing voice recording in the Anechoic Data is convolved with the impulse response to yield the reverberant input signal. The spectrogram is shown in Figure 1.

Then, the dereverberation algorithm is performed on a recording of symphonic music [3]. A short clip of a music piece by Mozart is separately convolved

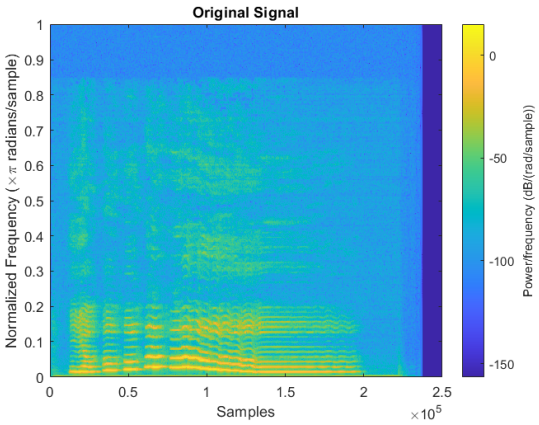


Figure 1: Spectrogram of Input Signal

with the two levels of impulse responses. For average reverberation, the impulse response is the one used in hyperparameter setting exploration. For high reverberation, the impulse response of Central Hall, University of York, is utilized.

3.2 Hyperparameter Setting Exploration

In this project, the algorithm is implemented in MatLab. First, the Short-Time Fourier Transform (STFT) is calculated to operate in a block-based manner. A Hanning window of 1024 samples with 75% overlap is applied to the input signal. At the end, the signal is converted to the time domain using the Overlap-Add (OLA) method.

As discussed above, the quality of the extracted dry signal is determined by six hyperparameters: B, γ, α, $MinGain$, $MaxValue$, and $Bias$. The hyperparameter setting table can be found in Appendix B and the spectrograms can be found in Appendix A.

B is the block size of the system. To get audible results, the minimum value of B is the hop size. From figure, it can be concluded that the amount of dereverberation increases as the value of B becomes larger.

γ is the smoothing vector for the gain vector. Figure 2(a) and Figure 2(b) shows the spectrograms with different γ values. It is shown that amount of reverb extraction is inversely proportional to the amount of smoothing. Besides, the reverb extract is mainly concentrated in the low-frequency part where the direct component ends.

α is the smoothing vector for the estimation of the system impulse response. Figure 2(a), 2(e), 2(f) and 2(g) show the spectrograms with different values of α. The amount of dereverberation is proportional to the value of α. In addition, the relationship is nonlinear. The amount of dereverberation increases dramatically when α is close to 1. Furthermore, with larger α, the dereverberation is no longer concentrated to the small portion. In contrast, the method can extract the reverberant component at the beginning of the recordings and in the higher frequency part.

$MinGain$ is a very small positive value close to zero. It makes sure that $G_S(\omega)$ will not be too small. Figure 2(h) and Figure 2(i) depict the spectrograms of $MinGain = 0$ and $MinGain = 0.05$. It shows that the amount of dereverberation decreases when $MinGain$ increases. This is probably because the smoothing process will fill up $G_S(\omega)$ at certain frequencies with a large $MinGain$ minimum value.

$MaxValue$ limits the range of the impulse response estimation. Figure 2(h) and Figure 2(j) show the spectrograms with different $MaxValue$s. It shows that the dereverberation amount is proportional to the value of $MaxValue$.

$Bias$ is a value greater than 1.0. It helps the estimation of the impulse response get rid of an incorrect minimum value. Figure 2(h), 2(k) and 2(l) show the spectrograms with different $Bias$ values. To get an audible result, $Bias$ can only float in a small range. It is shown that the amount of dereverberation is proportional to the value of $Bias$.

Based on subjective evaluation, the enhancement of dereverberation will aggravate the artificial effect. To get a natural result, the values of γ and α are important. In general, hyperparameter setting 2(h) yields the most natural result for this signal.

3.3 Implementation on Symphonic Music Signals

Table 2 shows the hyperparameter settings of the dereverberation system. It shows that a high value of $Bias$ is needed to extract a large amount of reverberation.

While tuning the hyperparameters, I noticed that to deal with the input signal with high reverberation, $Bias$ needs to be tuned greater. In addition, smaller values of γ and α is needed. In other words, high reverberation extraction needs a larger amount of gain vector smoothing, yet a smaller amount of impulse response estimation smoothing. The audio can be downloaded via this link: `https://github.com/yinanazhou/Dereverberation`. Based on subjective evaluation, this dereverberation algorithm does not work well with

high reverberation. Moreover, the artifacts mainly appears in the high-frequency part.

Table 1: Hyperparameter Settings for Symphonic Music Signal

Type	B	γ	α	$MinGain$	$MaxValue$	$Bias$
Average Reverb	Hop Size	0.2	0.2	0	0.99	1.015
High Reverb	Hop Size	0.05	0.08	0	0.99	1.29

4 Conclusion

This project implements the dereverberation algorithm proposed by Gilbert A. Soulodre [1].

First, the hyperparameter settings are explored on a singing voice recording. In general, the amount of dereverberation is proportional to the values of B, α, $MaxValue$ and $Bias$. However, it is oppositely proportional to the values of γ and $MinGain$. To get a natural result, the settings of γ and α are important.

In addition, t his algorithm is also implemented on a phrase of a symphonic music piece by Mozart. In general, this algorithm does not perform well on high reverberation. Moreover, high reverberation extraction needs a larger amount of gain vector smoothing, yet a smaller amount of impulse response estimation smoothing.

References

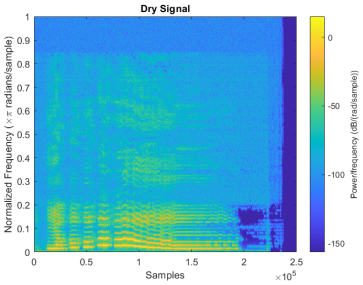
[1] G. A. Soulodre, "About this dereverberation business: A method for extracting reverberation from audio signals," in *Audio Engineering Society Convention 129*, Audio Engineering Society, 2010.

[2] D. T. Murphy and S. Shelley, "Openair: An interactive auralization web resource and database," in *Audio Engineering Society Convention 129*, Audio Engineering Society, 2010.

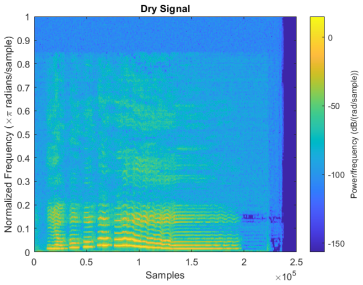
[3] J. Pätynen, V. Pulkki, and T. Lokki, "Anechoic recording system for symphony orchestra," *Acta Acustica united with Acustica*, vol. 94, no. 6, pp. 856–865, 2008.

Appendices

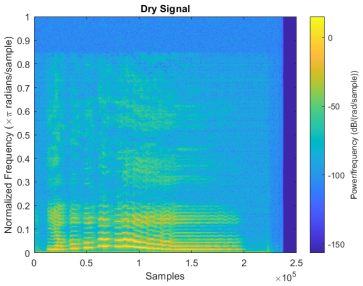
Appendix A Spectrogram Results



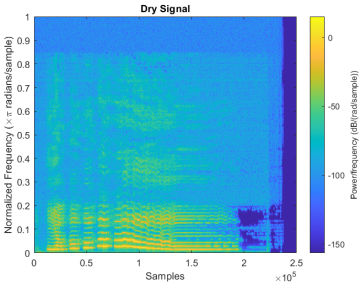
(a)



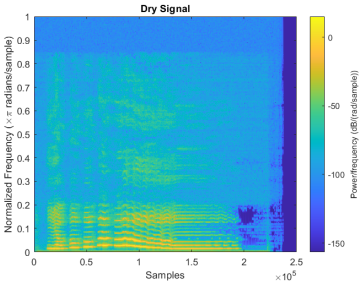
(b)



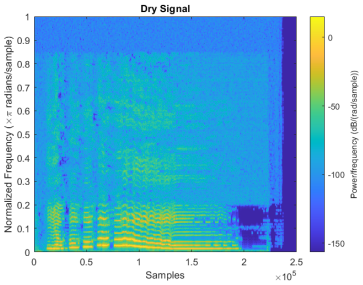
(c)



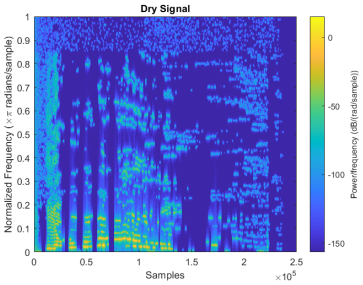
(d)



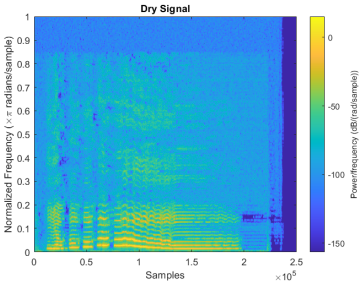
(e)



(f)



(g)



(h)

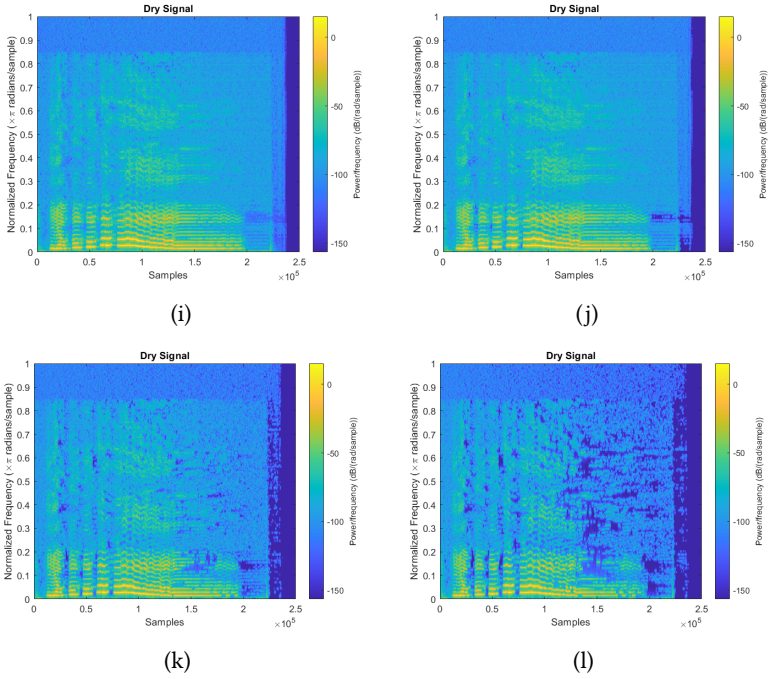


Figure 2: Spectrograms Results

Appendix B Hyperparameter Settings

Table 2: Hyperparameter Settings

Figure ID	B	γ	α	$MinGain$	$MaxValue$	$Bias$
a	400	0.99	0.5	0	0.99	1.01
b	Hop Size	0.99	0.5	0	0.99	1.01
c	400	0	0.5	0	0.99	1.01
d	400	1	0.5	0	0.99	1.01
e	400	1	0	0	0.99	1.01
f	400	1	0.8	0	0.99	1.01
g	400	1	1	0	0.99	1.01
h	Hop Size	1	0.8	0	0.99	1.01
i	Hop Size	1	0.8	0.05	0.99	1.01
j	Hop Size	1	0.8	0	0.2	1.01
k	Hop Size	1	0.8	0	0.99	1.05
l	Hop Size	1	0.8	0	0.99	1.1

Appendix C Links

- The code and audio samples of this project can be found via this link:

`https://github.com/yinanazhou/Dereverberation`

- The Open AIR Library are downloaded via this link:

`https://www.openair.hosted.york.ac.uk/`

- The symphonic music piece is downloaded from this website:

`https://research.cs.aalto.fi/acoustics/virtual-acoustics/research/acoustic-measurement-and-analysis/85-anechoic-recordings.html`