

Add Reverb to Audio with Convolution

Introduction

You can add special effects (such as echo and reverb) to audio with a technique known as *convolution*. Audio convolution needs two signals:

- the *dry audio* - this is sound recorded close to its source with no distortion, echo or reverb.
- the *impulse response* - this is a short burst of sound recorded in a specific physical space, such as an auditorium, kitchen, cave or even a shower cubicle.

You can use a balloon pop, a firecracker or a clap of the hands if you want to record an impulse response yourself. An ideal impulse is typically short, covers a wide range of frequencies, and can be consistently reproduced.

After convolution with the impulse response, the dry audio becomes *wet audio*. The wet audio has the frequency characteristics of the impulse response. This means the dry audio will sound as if it was recorded in the space the impulse response was recorded in.

Simply by changing the impulse response, you can make dry audio sound very different.

In this application, we:

- import a recording of a human voice, recorded on a microphone near to the speaker.
- import an impulse response (a single clap of the hands recorded in an enclosed space that has hard walls).
- convolve the audio with the impulse response.

After convolution, the human voice now has echo and reverb.

Play the audio with the  button at each stage of the application.

> **restart:**

Audio File

This is the dry audio

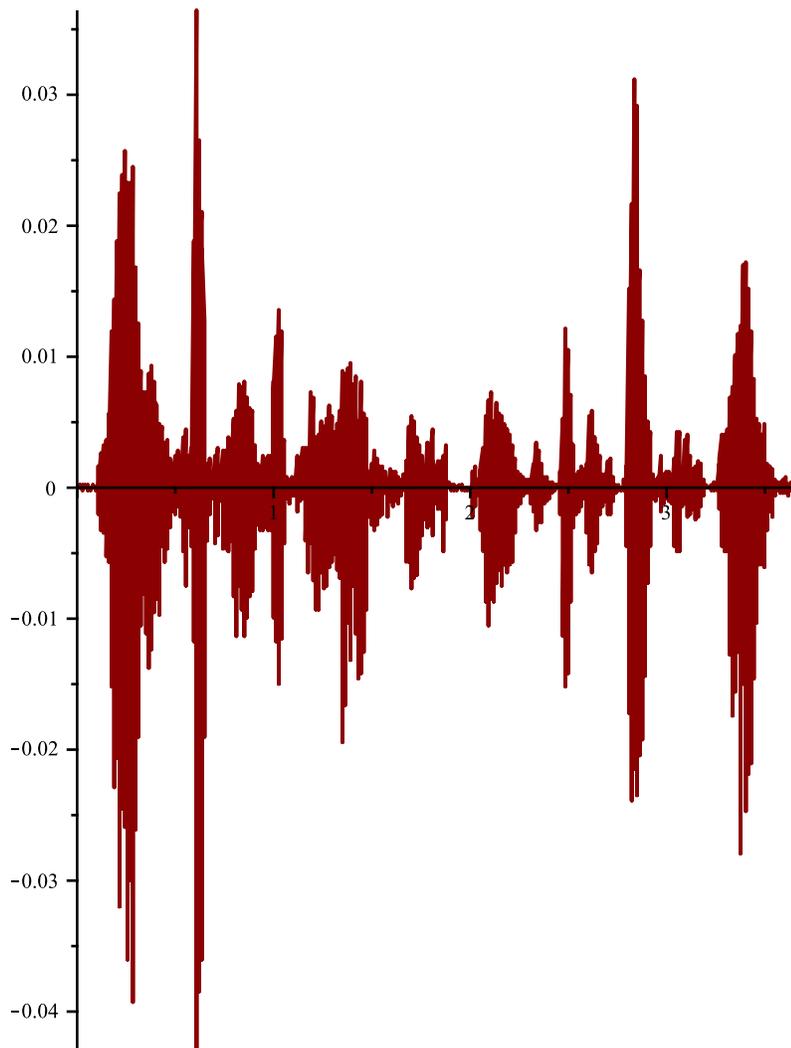
```
> clean_audio := AudioTools:-Read("this:///MaryHadALittleLamb_sc.
```

```
wav")
```

```
clean_audio := [ "Sample Rate"    44100  
                 "File Format"    PCM  
                 "File Bit Depth" 16  
                 "Channels"       1  
                 "Samples/Channel" 160768  
                 "Duration"       3.64553s ]
```

(2.1)

```
> AudioTools:-Preview(clean_audio, output = embed, color =  
  "DarkRed")
```



Impulse Response

This is the impulse response (or kernel) of the convolution.

The clean audio will be convolved against the impulse response

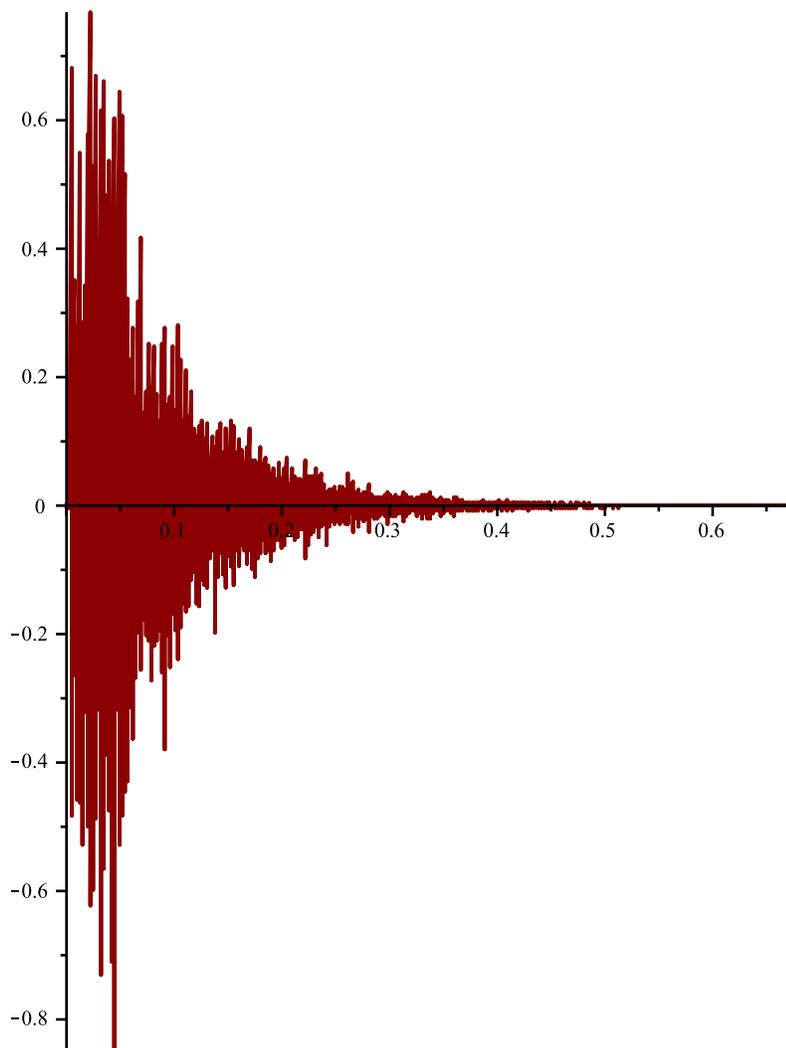
```
> impulse_response := AudioTools:-Read("this:///clap_sc.wav")
```

```
impulse_response :=
```

"Sample Rate"	48000
"File Format"	PCM
"File Bit Depth"	16
"Channels"	1
"Samples/Channel"	32407
"Duration"	0.67515 s

(3.1)

```
> AudioTools:-Preview(impulse_response, output = embed, color = "DarkRed")
```





Convolution

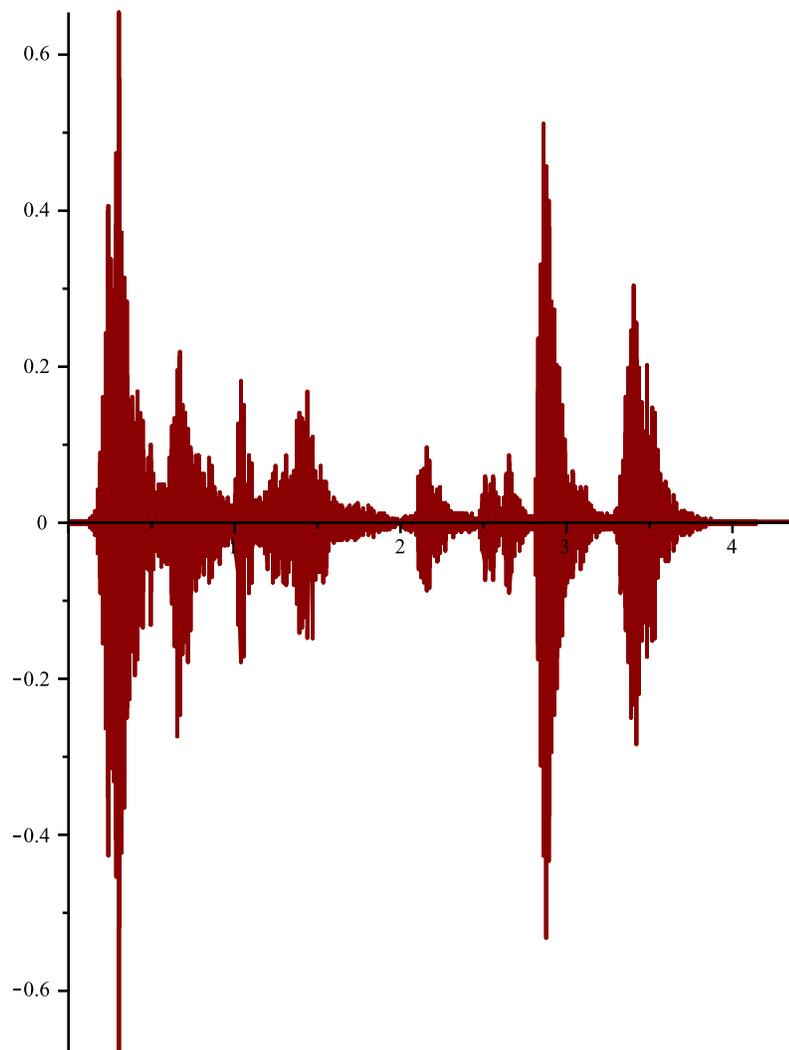
```
> tt := time[real]():  
   reverb_audio := AudioTools:-Convolution(clean_audio,  
   impulse_response);  
   time[real]()-tt;
```

```
reverb_audio := [ "Sample Rate"      44100  
                  "File Format"     PCM  
                  "File Bit Depth"  16  
                  "Channels"        1  
                  "Samples/Channel" 193174  
                  "Duration"        4.38036 s  
                  0.038 ]
```

(4.1)

This is what the audio sounds like with the sound signature of the enclosed space

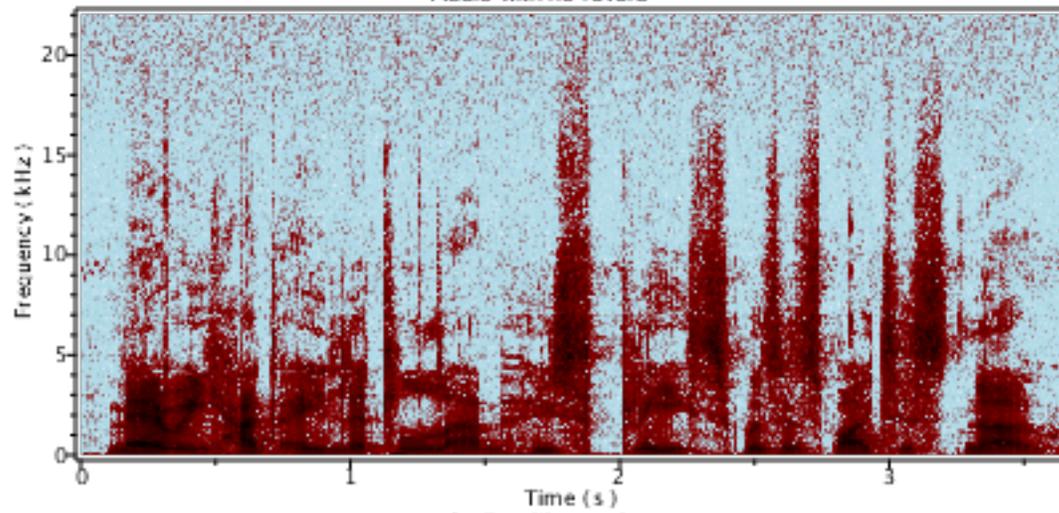
```
> AudioTools:-Preview(AudioTools:-Normalize(reverb_audio, amplitude  
= 0.85), output = embed, color = "DarkRed")
```



Time-Frequency Visualization with Spectrograms

```
> SignalProcessing:-Spectrogram(clean_audio, fftsize = 512, overlap = 0.5, colorscheme = ["zgradient", [white, LightBlue, DarkRed, black], markers = [0, 0.5, 0.6, 1]], size = [800, 400], title = "Audio with no reverb", titlefont = [Arial, 14]);  
SignalProcessing:-Spectrogram(reverb_audio, fftsize = 512, overlap = 0.5, colorscheme = ["zgradient", [white, LightBlue, DarkRed, black], markers = [0, 0.5, 0.6, 1]], size = [800, 400], title = "Audio with reverb", titlefont = [Arial, 14]);
```

Audio with no reverb



Audio with reverb

