

Can I do band filtering easily with DirectSound? If not, how can I do it?

Asked 16 years, 1 month ago Modified 15 years, 11 months ago

Viewed 2k times



I'm wondering whether something like this is possible (and relatively easy to do), and if so, how I could do it?

2



I would like to do band filtering on a wave file I'm reproducing. Something similar to the "Equalizer" you see in most Winamp-like applications.



My idea is, however, not to equalize the sound, but to use very high negative dB values, to almost kill the band I'm filtering.



The first question is: Does DirectSound give me something that allows me to do this?

If not: How would you go around implementing this?

I know (although I don't quite understand it completely) that you can convert from the sampled waveform to the distribution of frequencies using a Fast Fourier

Transform. Now, I obviously can't go back from that distribution to the original waveform after changing the amplitude values of certain frequencies :-)

How could I do something like this?

Also, how precise can I make these filters? (If I wanted to filter out everything from 2250Hz to 2275Hz, what would

be the error a filter would have? What would the maximum precision that I can get depend on?)

Thanks!

audio

filtering

signal-processing

waveform

equalizer

Share

Improve this question

Follow

asked Nov 3, 2008 at 13:51



Daniel Magliola

32.3k ● 61 ● 174 ● 245

7 Answers

Sorted by:

Highest score (default)



3



I don't know if DirectSound offers this functionality, I'd assume it doesn't since DSP is fairly complicated and often varies a great deal from situation to situation. What you want to do is typically called "filtering" in DSP (digital signal processing). Many times this involves using a FIR (finite impulse response) filter. There are many libraries out there to do exactly what you want. One of the trickiest aspects of filter design is that there are always tradeoffs between speed, accuracy, and error. In your example, you will be able to remove signal between frequencies but this will also effect the surrounding frequencies. The amount it will effect is related to processing time and filter design.

Perhaps start here (math heavy) : [FIR Filter](#)

Then google for your own Windows/DirectSound specific FIR related information

Share Improve this answer

answered Nov 3, 2008 at 14:01

Follow



basszero

30k ● 9 ● 58 ● 80



2



DirectSound does not do band filtering like you describe here, as far as I know.

The general idea behind bandpass filtering is to use delay lines, which take the signal output and feed it back into the input stream, with a specified delay time and decay (or attenuation) factor. Careful design of the filter will allow you to amplify or attenuate specific frequency ranges in your audio source. Note that this technique does not use FFT, except as maybe a diagnostic tool while testing filter effects. FFT techniques do a more precise job of limiting or amplifying frequencies, but delay lines are generally faster (and way easier to code).

For processing a WAV file (as opposed to doing real-time synthesis/filtering), executing a delay line on your audio buffer is as simple as:

```
for (int i = 0; i < samples.Length - delay; i++)  
{  
    samples[i + delay] += samples[i] * decay;  
}
```

It's a bit more complicated than this in practice (you have to deal with potential overflow values, for example, and some types of delay lines have to be run in reverse, which is always a pain in C-style coding), of course.

As far as how precise the filter is, that just depends on how well it's designed (it's very difficult). When you design a filter using delay lines, you are essentially doing the same thing that electrical engineers did (and still do) in the decades before cheap microprocessors.

Share Improve this answer

edited Nov 3, 2008 at 14:27

Follow

answered Nov 3, 2008 at 14:04



MusiGenesis

75.3k ● 41 ● 197 ● 338



1



DirectSound offers no signal processing facilities at all. There are a variety of techniques you could use to do what you want. It is possible to use the FFT to do what you want, but it's probably not the best or easiest method. You should read up on audio DSP, particularly digital filtering (IIR, FIR). There's a good DSP book available online for free called [The Scientist and Engineer's Guide to Digital Signal Processing](#) which is definitely worth a look. There are also many other good DSP books available from Amazon etc.

Share Improve this answer

answered Nov 3, 2008 at 14:00

Follow



Stu Mackellar

11.6k ● 1 ● 40 ● 59



I don't know of any libraries that take care of this sort of thing directly.

1



You can achieve what you want with Fourier transforms, implementations such as the [FFTW](#) do the calculating work for you but in my experience are pretty nasty to work with, and use prodigious quantities of memory, especially if you want to process longer bits of audio in one hit.



The basic idea for applying eq using FFT is this:

1. Get your audio. Audio is just a very long array of values (samples) which is the displacement of the speaker cone / over time.
2. Take the Fourier Transform of the audio (the library will do this, but you'll have to shunt the audio samples into the right format. This converts the time based samples into a frequency based representation - essentially this converts the signal to show the distribution of frequencies in the signal.
3. Divide the frequency distribution up - split the distribution into regions, each region will be a range of frequencies.
4. You can then perform adjustments on the frequency bands - for your example you could zero one region to remove all trace of it.

5. Take the inverse Fourier Transform of the updated frequency distribution. This will return the representation to the time domain, reconstructing an (approximation) of the original signal, but with the adjustments you have made.

Doing something like this will let you accurately manipulate the frequencies present in your audio giving you the kind of control you seem to want. Be warned however that it is not straightforward to implement.

I recommend reading around the subject. Beat detection is quite closely linked to a lot of this (uses the base techniques a lot) - try the first few sections [here](#) as a start.

Hope that helps a little.

Share Improve this answer

Follow

answered Nov 4, 2008 at 12:20



xan

7,568 ● 8 ● 46 ● 70



DirectSound might not support this directly, but DirectShow should. Could you use that API instead?

0



Share Improve this answer

Follow

answered Nov 14, 2008 at 19:44



AShelly

35.5k ● 16 ● 95 ● 157





I second the FIR filter idea. To get a narrow notch you will need a long filter kernel.

0



Basically you use convolution of the input stream against a set of values (the kernel). Every output sample is a sum of the previous N samples multiplied with its corresponding entry in the filter kernel.



So you need to keep a kernel array and a FIFO or circular buffer of the same number of samples.

There are FIR filter kernel calculators on the web, just google "FIR filter calculator".

Share Improve this answer

answered Jan 4, 2009 at 21:12

Follow



JeffV

54.4k ● 33 ● 105 ● 124



0



You should be able to do an FFT, mess around in the frequency-domain (scaling frequency bins), and then do an IFFT to recover the time-domain signal. Otherwise, designing filters using ScopeFIR or MATLAB is pretty easy. ScopeFIR can easily design a bandstop filter and give you the coefficients so you can do a convolution on the signal. Here's a tutorial from the ScopeFIR website:

<http://www.iowegian.com/fir/tutor/firintro.htm>

Share Improve this answer

answered Jan 28, 2009 at 5:57

Follow



switchmode

183 ● 4