Digital Audio Fundamentals: Spectrograms

Recall that we can view a signal in two ways: as a graph of amplitudes vs time (the Time Domain) or (because all periodic waves can be expressed as the sum of simple component waves) as a graph of frequencies vs amplitudes (the Frequency Domain):

By adding a third dimension (time) to a spectrum we get a spectrogram.

[Example: steelstring.wav]
Expressing these three dimensions graphically can be done with B&W intensity:

Digital Audio Fundamentals: Spectrograms

Expressing these three dimensions graphically can be done with B&W intensity, or color ("heatmaps")

Digital Audio Fundamentals: Spectrograms
Expressing these three dimensions graphically can be done with B&W intensity, or color ("heatmaps"), or faux 3D:
Digression for fun..... spectrogram artwork can be produced by various applications which take an image and create sounds that produce the given image......

https://www.youtube.com/watch?v=M9xMuPWAZW8

To capture the complexity of a real musical sound, you would have to account for the varying amplitude of each component sine wave individually, by giving the amplitude envelope function another argument for the frequency:

```python
def exponentialAHD( i, A, B, h, f):
    ....
```

For example, the decay rate could be modified depending on the frequency.

We will do a simple example of this in HW 03 (imitating the steel string note) and return to discuss it in more detail after we learn about the Fourier Transform (which can extract spectra from a sound).
Detecting the fundamental frequency of a signal ("pitch detection", more properly "F0 detection") is a good place to start in our study of musical signal analysis. For some signals it is easy to see their fundamental frequency:

\[
0.3305 - 0.326 = 0.0045 \\
1/0.0045 = 220 \text{ Hz}
\]

These signals have a clear sense of a dominant pitch.

Other sounds are harder to determine or have no pitch:

[Example: Beethoven's Seventh Symphony.]

[Higher love, percussion opening.]

Interestingly, although we are not always conscious of it, speech sounds do usually have a dominant pitch, although some consonants ("s") are practically pitchless.

[Genesis01.wav]
Other sounds are harder to determine or have no pitch:

[Beethoven’s Seventh Symphony.]

[Higher love, percussion opening.]

Interestingly, although we are not always conscious of it, speech sounds do usually have a dominant pitch, although some consonants (“s”) are practically pitchless.

[Genesis01.wav]

The simplest pitch detection is the Zero-Crossing Rate (ZCR) Algorithm; since a sine wave crosses the 0 line twice in its period, you could just count the number of times the signal changes sign (+ to – or – to +) in a signal window and divide by 2*length of the window:

\[ \frac{4}{2 \times 0.1s} = 20 \text{ Hz} \]
There are various problems with the approach, first, that it can be off by half a period:

\[
\frac{4}{2 \times 0.1 \text{s}} = 20 \text{ Hz} \quad \text{but the signal is 23.2 Hz}
\]
The second problem is that higher harmonics in the signal may create too many zero crossings!

The solution is typically to smooth the signal using a "low-pass filter" that filters out the higher harmonics. This is the approach taken by cheap "chromatic tuners for guitars".

A better method can be designed by using the notion of autocorrelation of a signal:

Pearson’s Product-Moment Coefficient for random variables $X$ and $Y$:

$$
\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},
$$

But for signals, there are several simplifying conditions:

1. Since we will be interested only in comparing magnitudes (e.g., finding the maximum), we do not need to normalize to the range [-1..1], and so can eliminate the denominator;
2. Since signals have a mean very close to 0, we do not need to standardize by subtracting the means (already standardized!)

So we would have $E(X \cdot Y) = \Sigma(x_i \cdot y_i) / \text{len}(X)$, but, again, as in (1), we do not need the denominator! We have simply: $\Sigma(x_i \cdot y_i)$
Example: Suppose X and Y are identical: All the products are positive, because the negatives cancel (sum will be large):

Example 2: Suppose X and Y are identical except phase shifted by $\pi$: All the products are negative (sum will be small):
Example 3: Suppose X and Y are identical except phase shifted by $\pi/2$:
Some products positive, some negative, sum will be 0.

In fact, the correlation varies as a sine wave, with a maximum at 0 and $2\pi$ and a minimum at $\pi$, and 0 at $\pi/2$ and $3\pi/2$ (effectively a cosine):
Autocorrelation is the same thing as correlation, but the Y signal is a time-shifted version of the X signal:

$$\Sigma (x_i \times x_{i+\text{lag}})$$