Worksheet – Note Detect

Prerequisites:

This worksheet assumes that you have already completed the wave player and recorder tutorials.

Goals:

- To determine the notes that a person sings.

Introduction:

Even if you didn't do my midi sampler tutorial, you probably know that every pitch has a fundamental frequency, these are listed on the last page. Now this tutorial assumes you are using wave files (though any uncompressed pcm format will work), so your raw data is in the time domain. You have a series of numbers each representing the amplitude at the next time interval.

To get at the pitch, you will first need to convert your data to the frequency domain. This can be done with an Fast Fourier Transform (FFT), but before you go rushing off to do so there are some things that must be considered. The more data you feed into FFT, the greater the resolution of your frequency domain output. However, you lose time resolution, so if more than one note is sung you will want to feed the data in in chunks. I recommend using files with a sample rate of 44.1 kHz and then using chunks 16,384 samples long (must be a power of two for the most efficient FFT's). This will give you a couple of notes per second and allow for a frequency resolution of 2.75 Hz, (which is important for lower octaves). Remember by the Nyquist Theorem, you get frequencies up to ½ of your sample rate; meaning 22.5 kHz for 44.1 kHz. Also, when you feed data into an FFT, you get the same number of complex return values as you gave it, but only the first half are useful. This means that 16,384 samples will give you 8,192 frequency bands. These bands represent the amplitude of all the frequencies in their domain.

In a pure tone, the frequency band with the highest magnitude contains the frequency of the pure tones pitch. To get the note, simply find the average frequency of the largest frequency band, and find the note that is closest to it (usually contained in the band for pure tones).

However, a person's voice is not a pure tone, and contains lots of different component frequencies. The bad news is the maximum frequency does not have to be the pitch of the person's voice. Now for the good news, the maximum frequency should be a harmonic (or an integer divisor) of the fundamental frequency, and the fundamental frequency will be larger than those around it. To make this task a little easier below is an algorithm for finding the fundamental frequency:

– First find the five maximum frequency bands ensuring they are all at least 5% different from each other to prevent a cluster of points around one harmonic.
– Second arrange the points by frequency (highest to lowest)
– Finally find the lowest frequency that is an integer divisor of the highest frequency (within a certain margin of error, I suggest 5%).

This should yield the frequency band containing the fundamental frequency. Depending on how high the fundamental frequency is, it may or may not be in the same band as the pure pitch it represents, so find the closest pure tone (or note) and return it.

Tasks:

- Ensure that you have a sound file handler of some sort coded
- Get a pre-coded FFT and learn how to implement it
- Use the algorithm in the introduction to make a main file that finds which note a person sings

Additional Information

Note Detect, last revised on 07/24/08

Note	Frequency	Note	Frequency
C2	65.41	C3	130.81
C2#	69.3	C3#	138.59
D2	73.42	D3	146.83
D2#	77.78	D3#	155.56
E2	82.41	E3	164.81
F2	87.31	F3	174.61
F2#	92.5	F3#	185
G2	98	G3	196
G2#	103.83	G3#	207.65
A2	110	A3	220
A2#	116.54	A3#	233.08
B2	123.47	B3	246.94
C4	261.63	C5	523.25
C4#	277.18	C5#	554.37
D4	293.66	D5	587.33
D4#	311.13	D5#	622.25
E4	329.63	E5	659.26
F4	349.23	F5	698.46
F4#	369.99	F5#	739.99
G4	392	G5	783.99
G4#	415.3	G5#	830.61
A	440	A5	880
A4#	466.16	A5#	932.33
B4	493.88	B5	987.77
C6	1046.5		
C6#	1108.73		
D6	1174.66		
D6#	1244.51		
E6	1318.51		
F6	1396.91		
F6#	1479.98		
G6	1567.98		
G2#	1661.22		
A2	1760		
A2#	1864.66		
B2	1975.53		