

Visualization of Musical Pitch

Philip McLeod, Geoff Wyvill
Department of Computer Science
University of Otago, New Zealand
Email: pmcleod@cs.otago.ac.nz

Abstract

We have created software that shows a musician the pitch of the notes he or she is playing or singing, in real time and very accurately. This is useful as a teaching aid for beginners and also for studying refinements of sound production such as vibrato.

Keywords: *User interface, pitch determination, visualisation, real time, Fourier transform, harmonics.*

1 Introduction

Music is the most abstract form of art. Sound appeals to us emotionally and very directly. And, as in every art form, this seemingly effortless communication is achieved by an intensity of study and attention to detail that almost defies belief. Serious musicians spend countless hours refining technique and eliminating tiny defects that most of their audience would never be aware of at a conscious level.

Our long term goal is to develop a variety of tools with which to study all aspects of musical performance. A graphical interface can give immediate, non-verbal and accurate feedback to a musician. It can also help us to analyse performance and learn something about how we make artistic and technical choices.

We have started in a small way with a study of musical pitch. That is to say we are looking at how high or low the frequency of a note is without regard to the timbre or tone quality. A note can be “in tune” or “out of tune” but within our idea of what is “in tune”, the artist can make choices. Singers, and those who play instruments that permit such variation, consciously or unconsciously, pitch their notes subtly differently in different contexts to produce different artistic or emotional effects.

The experimental tool we have created enables beginners to learn to find the right notes accurately and more advanced players to study these subtler variations with the help of an independent and objective tool. One of the things

a performer needs to do is listen in a critical frame of mind. An actor in comedy does not laugh because he or she must maintain the character. A musician may enjoy his or her own playing but must maintain concentration on what is to be improved. We do not always know why something sounds better or worse. This knowledge can help us to improve our art and help us to know where to concentrate our efforts.

The great physicist, Helmholtz, studied the intonation of the famous violinist, Joseph Joachim, and reported that he played an unaccompanied scale in “just and not even tempered intervals” [9]. Even this simple conclusion was no mean feat in the 19th Century. Helmholtz had devised a set of resonators that enabled him to identify pitch reasonably objectively but it must have taken quite a time to observe the whole scale, checking each note individually with several resonators. And the resonators themselves each had to be calibrated against a standard siren that could be operated to produce sound at any given pitch.

Technology has made this kind of measurement easier and there are “electronic tuners” available commercially that will tell you if a note is above or below one of a number of fixed pitches. Note that this is a much simpler problem than determining the precise pitch of a given sound.

From our point of view, there are two main issues. We must determine the pitch of each note fast enough for immediate feedback to the musician, and we must display this information in an immediately useful form. In practice, this means we have to find the frequency of a signal within about 1/20 seconds and display the result as a continuously varying graph so the immediate history of the sound is also visible.

The determination of pitch is actually more than the determination of frequency. In order to know the pitch, it is necessary to deduce the dominant frequency from a number of frequency components of the wave. Our algorithms yield all these harmonic components and their amplitudes so we also have the basis for further analysis of timbre.

In the rest of this paper, we describe how we find the pitch with sufficient speed and accuracy. We then report

on our results with demonstration output and finally our experience testing the system with the help of a professional violinist.

2 Finding the pitch

The fundamental frequency of a note is the number of complete cycles of the waveform repeated every second. In principle, this can be determined from a Discrete Fourier Transform (DFT) applied to a set of samples of the waveform amplitude. In practice, the set of samples is taken over a short time interval that we call the "sampling window". To get a reasonably accurate result, you either need a very long sampling period (large window) or you need to guarantee that you have a whole number of cycles within the window. The large window is not possible if the note keeps changing and a whole number of samples implies that you already know the pitch.

Our approach is to multiply the samples within the window by a Gaussian function and use a Fast Fourier Transform (FFT) to find an approximation to the fundamental. It is then possible to generate a more accurate spectrum by smoothing and resampling the wave to create a window that contains a whole number of waves.

Algorithm outline:

1. Select a sampling window from the incoming data. For continuous display, these windows may overlap in time (section 2.1).
2. Apply the Gaussian function to the window (section 2.2).
3. Perform the FFT (section 2.3).
4. Identify principal frequencies (section 2.4).
5. Identify the fundamental as a sub-multiple of the frequency of greatest amplitude (section 2.5).
6. Recognise the fundamental as a note of the musical scale (section 2.6).

At this stage the pitch can be displayed. We also have enough information to smooth and resample the original data and repeat the pitch determination on a window that is a multiple of the fundamental frequency (section 2.7). This enables us to find the amplitudes of the harmonics accurately. It also improves the accuracy of the fundamental determination.

2.1 Window Selection

The FFT process, in practice, requires a number of samples that is a power of two. So at 44100 samples per second

we might choose 2048 as the window size because this is approximately 1/20 seconds and represents the rate at which we need the result.

2.2 Applying a Windowing Function

Applying a windowing function tapers the data at the window's ends, reducing discontinuities at its edges. The window's data is multiplied by the windowing function [5] [7]. This localises the time as well as the frequency when a DFT is performed on it, and it minimises "spectral distortion". Multiplying in the time domain is equivalent to a convolution in the frequency domain. By choosing a suitable windowing function, we can arrange that the convolution achieved reduces spectral leakage effects [6]. A Gaussian windowing function is used for reasons described in section 2.3.

2.3 Performing an FFT on Windowed Data

Using a straight FFT (with no windowing function) generates the narrowest possible main lobe in frequency space, but large sinc-function side lobes. No spreading of exact harmonic components occurs; but it produces lots of spectral leakage with non-harmonics. Spectral leakage is loss of information about the exact frequencies, being spread across the integer multiples of the transform (figure 1). Using a windowing function reduces the size of the side lobes, but at the same time broadens the main lobe of the spectral line.

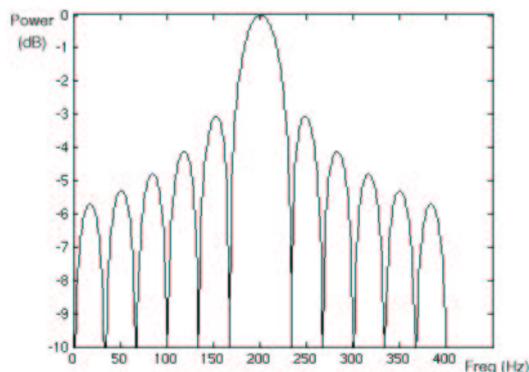


Figure 1. Here a single frequency has been spread by spectral leakage, with a large main lobe and many side lobes.

There is a fundamental limit on what windowing functions can optimise. This is defined by the Uncertainty Principle [10]. The function which meets the bound, giving

the smallest side lobes, is the Gaussian windowing function (figure 2) [4]. The use of the Gaussian windowing function also results in nearly phase invariant spectral coefficients. The Gaussian window function used is

$$y(x) = e^{-\frac{1}{2}(2.5(\frac{2x-w}{w}))^2} \quad (1)$$

where w is an even window width, and x runs from 0 to $w - 1$.

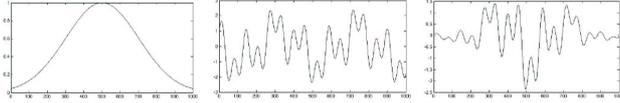


Figure 2. The Gaussian windowing function, a wave, and the Gaussian applied to the wave.

2.4 Finding Gaussian Peaks

Multiplying the original wave by a Gaussian has the same effect as convolving the Fourier spectrum with a Gaussian. So, Gaussian smoothing has been performed in the frequency space. We now have a set of frequency coefficients that are values from the underlying continuous function made up of the addition of Gaussians. The assumption is made that these Gaussian shapes from each harmonic are far enough apart that we can treat each one locally as just one Gaussian (Figure 3). The maximum of each of the localised Gaussians can then be found.

A local centre can be found by using three of the coefficients about it, fitting a Gaussian equation to them and finding the mean.

The general form for a Gaussian equation is

$$y = he^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

Where h is the height, σ the standard deviation, and μ the mean. By taking the logarithm of both sides of the equation,



Figure 3. The dots representing the coefficient values on the underlying sum of Gaussian's function can be used to find the local centre frequencies.

this is reduced to a quadratic that can easily be solved:

$$\log_e(y) = \log_e(h) + \frac{(x-\mu)^2}{2\sigma^2} \quad (3)$$

$$z = ax^2 + bx + c \quad (4)$$

where

$$z = \log_e(y) \quad (5)$$

$$a = \frac{-1}{2\sigma^2} \quad (6)$$

$$b = \frac{\mu}{\sigma^2} \quad (7)$$

$$c = \log_e(h) - \frac{\mu^2}{2\sigma^2} \quad (8)$$

where h , μ and σ are constant for a particular solution. If we shift the axis so it is centred on the middle of the three points, we can define the intermediate variables z_{-1} , z_0 , and z_1 as follows

$$z_{-1} = a - b + c \quad (9)$$

$$z_0 = c \quad (10)$$

$$z_1 = a + b + c \quad (11)$$

The maximum can be found by equating the derivative to zero. This gives

$$x = \frac{z_{-1} - z_1}{2(z_1 + z_{-1} - 2z_0)} \quad (12)$$

the frequency at the maximum, with respect to this coordinate system.

2.5 Choosing the Fundamental

After the frequencies of all the local maxima have been found, their amplitudes are calculated. The strongest frequency is selected and assumed to be one of the harmonics from the note being played. The fundamental is not always the strongest frequency in a note. Figure 4 shows an example of a note where the fifth harmonic is the most powerful.

The pitch of the note is related to the ‘‘dominant’’ frequency. In this case, dominant means what a musician perceives to be the main frequency. This is not necessarily the lowest or the one with the largest amplitude. The following process appears to agree with our subjective observation almost all of the time.

Firstly we pick out the frequency, f , with the largest amplitude. This is almost certainly a true harmonic of the fundamental we are seeking. In other words, the fundamental frequency is $F = f/n$ where n is an integer. For each value of n from one to ten we examine the spectrum to see how many frequencies are potential harmonics of F . A frequency

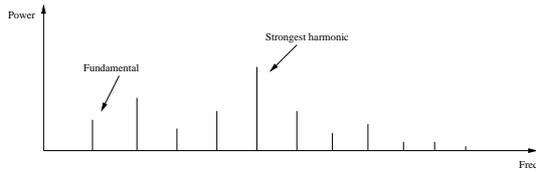


Figure 4. An example of a harmonic note and the power associated with each frequency. Here the fifth harmonic is the strongest.

is a potential harmonic if it is close the ideal frequency of the harmonic. We actually calculate a score for each $F = f/n$ which is a weighted sum of closeness measures. The more peaks that are separated by F the better the fit.

We have tested various measures of closeness and their combinations. For example, if the difference of two potential harmonics is very close to F , this is good evidence that F is the correct fundamental even if the potential harmonics are not exact multiples of F . We are still refining it.

Basically this process works like a voting scheme. The “frequency differences between peaks” vote for how well they fit each of the proposed fundamental frequencies (using Gaussian drop off). The strength of their votes is based on the square of the constituent amplitudes (in dB). The votes are then normalised by dividing by the total number of possible votes. The fundamental with the highest number of votes is chosen as the pitch of the note. The output from this stage is a good approximation to the pitch of the note which can be displayed.

2.6 Calculating Which Note on the Scale

Once we have the fundamental frequency, it can be converted into a note on the western musical scale. Using the tempered scale [2], which has all the notes evenly spaced (on a log scale), the following equation can be used

$$n = \frac{\log(\frac{f}{27.5})}{\log(\sqrt[12]{2})} \quad (13)$$

where n is the number of semitones from $A0$, the lowest note of a piano. The decimal part of the number can be used to determine how sharp or flat from the tempered scale the note is. To convert this value into *cents*, a unit of sharpness, simply multiply the decimal part by 100.

2.7 Refining the Window

We cannot apply the FFT to this new window because it doesn’t have a power of two as the number of samples. So we smooth the input either by linear interpolation or with a

cubic B-spline [3] and resample the result. We then apply the Gaussian window and FFT to the new sample set. All the harmonics have periods that fit or almost fit a whole number of times into the new window. Thus the amplitude information for the harmonics is more accurately represented than from the calculation in the original window.

We use the FFT, of course, because it operates in $O(n \log(n))$ time. This resampling method appears to be better than the common approach of zero-padding the data to fit the specially sized window. Some quantisation noise gets introduced but there is no associated frequency shifting. Also, we can make the window any length. With zero-padding, it is still a whole number multiplied by the time between consecutive original samples.

We seem to get very similar results with linear interpolation and with the B-spline. The B-spline is a little slower but the smoothing operation is still $O(n)$. The total time complexity of the smoothing, resampling and FFT is still $O(n \log(n))$.

3 Displaying the Pitch

We implemented a musical pitch viewing program in C++ with the Qt [11] library using the techniques described in this paper. You sing or play an instrument into a microphone and the note being played is displayed in real time (about 1/20th of second delay), taking 44100 samples per second input. This was running on a Pentium III - 550MHz, with 256MB memory and a 16 bit sound card.

Figure 5 shows a screen shot of the pitch viewer program. It has four main views. Firstly, the top left view displays the raw sound wave. The bottom left displays the frequency of the note being played through time, with reference lines displaying the tempered scale. The bottom right displays the note being played on a musical staff. The top right shows the power of each of the harmonics in the note (on a logarithmic scale).

The program uses one window size in the first instance to approximate the fundamental frequency, and overlaps each window with half of the next window. Bigger window sizes result in less responsive but more accurate information, allowing for better detection of low notes. Smaller window sizes allow for faster response with less accurate information. If two notes are played very rapidly, a large window will cause an intermediate pitch to be displayed.

Multiple levels of window size could be used simultaneously, if you need both accuracy and fast response.

4 Results

Kevin Lefohn, executant lecturer in violin and viola at The University of Otago, gave us feedback on the program.

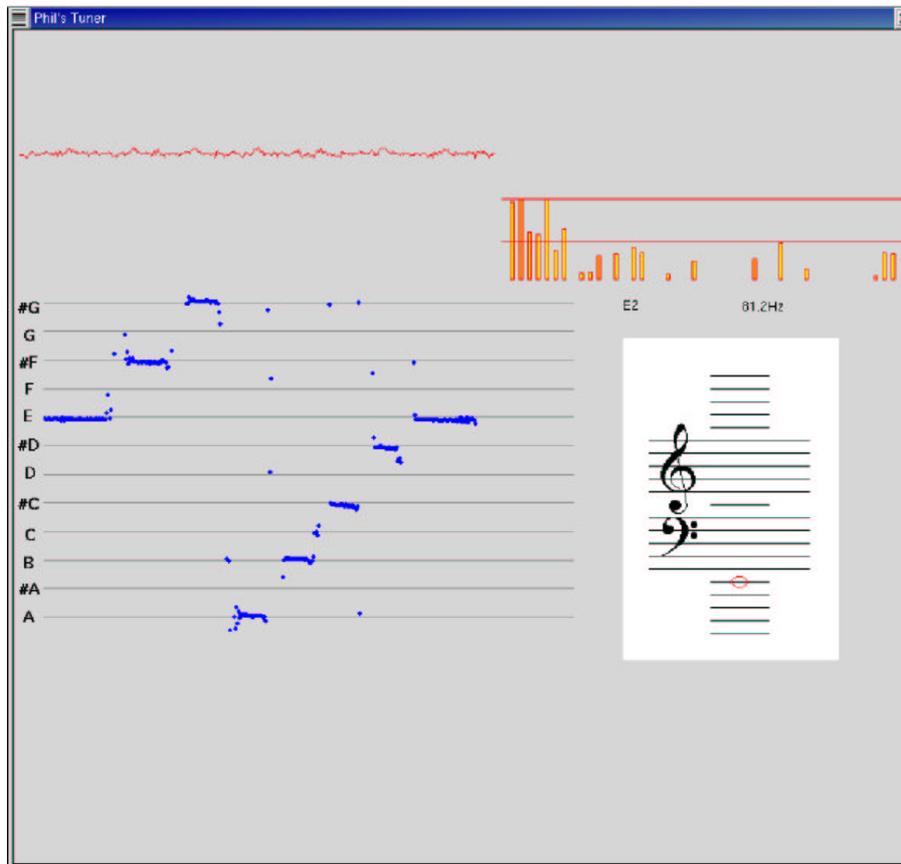


Figure 5. A screen shot of the pitch viewing program. Here an E major scale has been played on a guitar, with the last note E2 currently being played.

His comments were very encouraging and are incorporated in our findings below:

- The instant response is useful for visualising in real time where the pitch of the note being played is placed. People who play fretless instruments such as the violin, play subjectively. The musician can only determine the pitch of the note by ear. This skill can take many years to learn, and people’s opinions of pitch can vary. This program provides them with objective information about what is actually being played. They can see exactly how flat or sharp a note is.
- The program allows you to play a recorded sound over and over, becoming a learning tool for students or players wishing to have an “outsider” look/listen at what they have played. It allows experts to analyse what they have played at a later stage.
- One of the more fascinating discoveries that we made was the fact that the program picked up any discrepancy

in tone as a pitch changed, showing wobbles in the bow. It picked up little details such as slides between notes which we had never noticed before, but when we knew to listen for them we could hear them. Another detail was that some of the pitches he played on the way down a scale were slightly different from the notes he played on the way up the scale. He was doing this by choice because it gave him the sound he wanted. The program gives us a clear way to see and measure these discrepancies.

- Vibrato, a technique used to beautify a note by fast variation of the pitch of a note could be visualised clearly by the program. Typically vibrato oscillates around 4 to 5 times a second, and in theory the pitch is supposed to be on and below the specified note. But the program showed that this was not always the case. When the vibrato note sounded most in tune, the highest pitch was *above* the subjectively correct note. This was only tested on a very small sample set, so there is

room for more investigation here.

5 Discussion

The algorithm presented here seems to work well but is still in the early stages of development. Other methods can be tried for the interpolation and for choosing the fundamental. We can determine the amplitude of each harmonic in the note and this will lead to developments such as

- The harmonic frequencies in notes vary over time, giving notes their characteristic sound. Each harmonic creates an envelope over time describing the energy associated with it. These envelopes could be used to present information about the instrument and how it is being played. Things such as the way a guitar has been plucked affect these harmonic envelopes and could be studied more to understand them, and even reproduce the sounds mathematically.
- Not all harmonics in a note are a perfect multiple of the fundamental because things such as guitar strings are not perfect resonators. It would be interesting to investigate how sharp or flat each harmonic is with respect to the fundamental, and to see how much variation occurs.
- The pitch of a note appears to change as it gets louder. Is this actually a characteristic of the sound or is it an artefact of our perception? Some experiments could be done to investigate this.
- There are some automatic score writers available on the market, such as Sibelius¹ and Finale², but they do not go to sub-note accuracy. There is room for extending the MIDI format to include more information about how a note has been played, because in real music the same note is not played exactly the same every time.
- We should also study multiple notes played simultaneously. This is a lot more complicated but it is necessary if we are to study harmony.

6 Conclusion

We have presented a method for finding the pitch of a note in real time, with the ability to extract information about the power associated with each harmonic. It can calculate the fundamental frequency and give a response in about 1/20 of a second.

The pitch is displayed visually providing a useful tool for musicians. We have demonstrated the ability to analyse

¹www.sibelius.com

²www.codamusic.com/finale

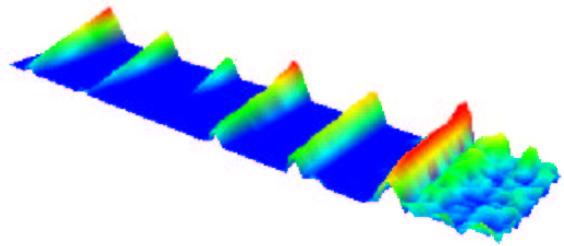


Figure 6. Analysis of a note over time showing the envelope of each harmonic component as a ridge.

vibrato. It can be used as a teaching tool, allowing teachers and students alike to visualise details of music as it is being played. It also makes a great tuning device which can be used by anyone, being both more responsive and easier to use than existing electronic tuners.

References

- [1] Wavelet analysis: Theory and implementation, 2002. URL <http://www.cs.ualberta.ca/~minghong/wavelet.doc>.
- [2] J. Murray Barbour. *Tuning and Temperament*. DA CAPO Press, 1972.
- [3] Peter Burger and Duncan Gillies. *Interactive Computer Graphics*. Addison-Wesley Publishing Company, 1989.
- [4] Y T Chan. *Wavelet Basics*. Kluwer Academic Publishers, 1995.
- [5] Richard C Dorf, editor. *The Electrical Engineering Handbook*. CRC Press, 1993.
- [6] Paul A Lynn and Wolfgang Fuerst. *Introductory Digital Signal Processing With Computer Applications*. John Wiley and Sons, 1998.
- [7] Mathworks. Signal processing toolbox, 2002. URL <http://www.mathworks.com/access/helpdesk/help/toolbox/signal/signal.shtml>.
- [8] Boaz Porat. *A Course in Digital Signal Processing*. John Wiley and Sons, 1997.
- [9] Percy A. Scholes. *The Oxford Companion to Music*. OUP, ninth edition edition, 1955, revised 1965.

- [10] Raymond Serway. *Physics For Scientists And Engineers With Modern Physics*. Saunders College Publishing, 4th edition, 1996.
- [11] Trolltech. Creators of qt, the cross-platform c++ gui toolkit, 2002. URL <http://www.trolltech.com>.