HARVEY MUDD COLLEGE

Computer Science Department

Using Wavelets for Monophonic Pitch Detection

John McCullough
Harvey Mudd College

September, 2005

HMC-CS-2005-01

Abstract

Wavelets are an emerging method in the field of signal processing and frequency estimation. Compared to Fourier and autocorrelation methods, wavelets have a number of advantages. Wavelets based on the derivative of a smoothing function emphasize points of change in a signal. Positive results were achieved when applying such a wavelet to estimate the fundamental frequency and corresponding pitch on the equal-temperament harmonic scale.

Keywords: Wavelets, Pitch Detection, Pitch Estimation, Fundamental Frequency Estimation

1 Introduction

At the start of the semester I was fascinated with the idea of automatically extracting structural information from music. I had recently seen a presentation on the use of machine learning methods to track individuals in the video from security cameras to assist in identifying deviant behavior (Jones). The algorithms they were using seemed promising and they were making good progress in addressing issues that arose in tracking an object in a scene and, at a symbolic level, could do very well. The major issues that exist in computer vision include: Identifying objects in the scene, identifying a moving object as the same between frames, identifying occluding objects within a scene, as well as issues with poor video quality.

During my initial fascination with structure in music, I ignored the issues associated with frequency estimation in real signals and superficially thought about the issues of pattern and movement in music at the symbolic level, in a manner similar to the surveillance video. Within machine learning, there are methods, such as n-grams, that can be applied to sequence prediction although none are able to solve the problem entirely. Then I began to wonder how we might extract the symbols from a real signal. This, like any type of machine perception, often seems straight-forward at first. We, as humans, are often very good at it. Unfortunately, transferring our abilities to a machine is very difficult, in part justifying the existence of the machine learning field.

2 Fundamental Frequency Transcription Techniques

A lot of work has been done in the field of musical transcription and computer music in general. A variety of methods have been devised to approach a number of them. I hope to address the fundamental ideas of a few that are applicable to monophonic pitch detection to further my own knowledge and to facilitate further research. There are several works that cover additional methods in significant detail, (1), and others that focus on more specific methods, (2).

2.1 Fourier

One of the most basic and ubiquitous methods in signal processing is the Fourier Transform (FT). It transforms a signal from time space into frequency space. For example, we could decompose a square wave (Figure 1) into its component parts. There are some assumptions inherent in the way the transform operates that are detrimental to identifying frequencies in time: if a frequency is present in a signal then it is present for the entire duration of that signal and that signal uniformly repeats from $-\infty$ to $+\infty$. The Discrete Fourier Transform (DFT) takes a signal, z, with N samples and calculates the coefficients, Z, corresponding to equally spaced frequencies starting with period equal to the length of the signal and ending with the sampling frequency, f_s:

$$Z(m) = \sum_{n=0}^{N} z(n) e^{\frac{(-2\pi i)(m)(n)}{N}} \tag{1}$$

where $N \in \mathbb{Z}^+$ and $m, n \in \{0, 1, 2, \ldots, N\}$. The $Z(m)$ coefficient corresponds to the frequency $(m \cdot f_s)$ (4).

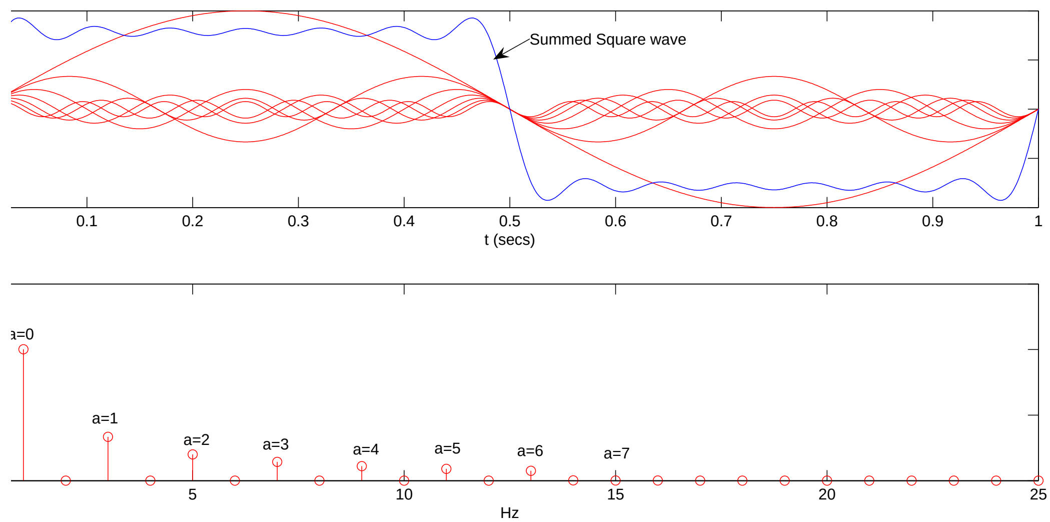


Figure 1: A Square wave is made by the sequence $f(t) = sin(2\pi t) + \frac{1}{3}sin(2\pi 3t) + \frac{1}{5}sin(2\pi 5t) + \frac{1}{2a+1}sin(2\pi(2a+1)t)$, $a \in \mathbb{N}$. In this example the first seven elements of this sum are included (Top). The first twenty-five DFT coefficients of this square wave are also shown (Bottom).

In the example square wave, we can look at the coefficients of the DFT and see that the appropriate frequencies are in the signal. However, this is only effective when the frequencies are evenly distributed across the signal. If we look at an example that is un-evenly distributed in time and frequency (Figure 2), we can see that there are peaks for the the frequency components of the signal at ± 16Hz and ± 3Hz, but they are not isolated. The presence of the other frequencies in this spectrum are necessary to flatten out the quiet parts. If we constructed a signal based on these coefficients on the interval 0s to 12s, there would be two exact repetitions of the signal shown at the top of Figure 2. If we look at the highest peaks we can identify what the fundamental components are, though this is not always the case. Because we are in the frequency domain, we cannot identify where in time these frequencies are present based solely on the spectrum. If we could pick out the frequencies in a signal, it would not do us much good just to know that they existed at some point in a long signal. It would be more useful if we looked at shorter sections of a long signal, because using a FT on the shorter sections, we can identify which frequencies exist at the time and for the duration of each short section.

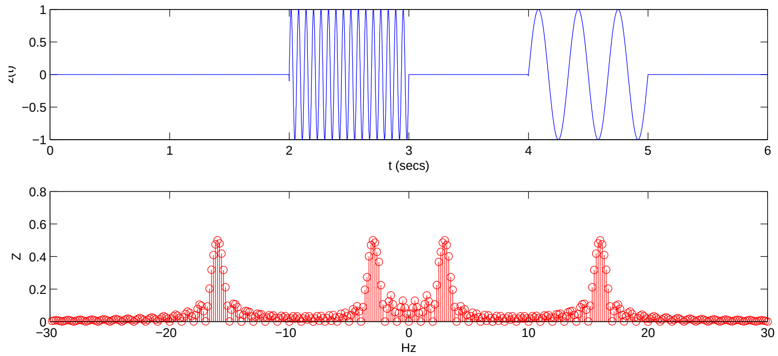


Figure 2: A 16Hz sine wave on the interval 2-3s and a 3Hz sine wave on the interval 4-5s (Top). The FT Spectrum (Bottom) of the signal indicates the presence of these two frequencies as well as a number of others.

The Gabor Transform, also known as the Short Time Fourier Transform (STFT), uses a windowed method. By sliding a window of a given size along the signal, we can use the FT to try to identify the frequencies present in each window. The temporal position of each window gives the temporal position of the frequencies in that window of the signal. By the nature of the FT, the length of the window determines the lowest detectable frequency and is the inverse of the window length (Figure 3). A longer window length lowers the detectable frequency range, but it also lowers the time resolution of the transform (5, 24). Thus, if we want to identify lower frequencies in a signal we reduce our ability to accurately pick out where any frequency occurs within the signal.

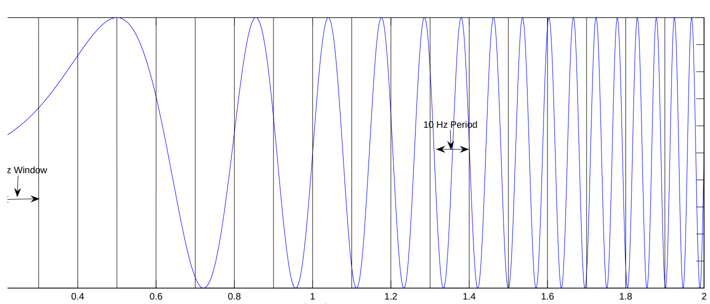


Figure 3: A sine wave with increasing frequency split into 10Hz windows. A single period of the wave does not fit into the window until the frequency passes above 10Hz.

2.2 Autocorrelation

Although, for frequency estimation the STFT has many advantages over the regular FT, other methods have also been applied to pitch detection. One such method is autocorrelation, which is achieved by convolving the complex conjugate[1] of a signal with itself. In the continuous case, this can be computed by (1):

$$R_x(\tau) = \int_{-\infty}^{\infty} x^*(t)x(t+\tau)dt \quad (2)$$

If the R values are computed over the length of the signal and the signal is sufficiently self similar, they should exhibit the same fundamental frequency as the signal (3) (Figure 4). In cases when the signal is not particularly self similar over its entire length (Figure 5) windowing methods can be used, although this can introduce significant computational costs if the windowing function is more involved than simply analyzing a fixed number of samples (3).

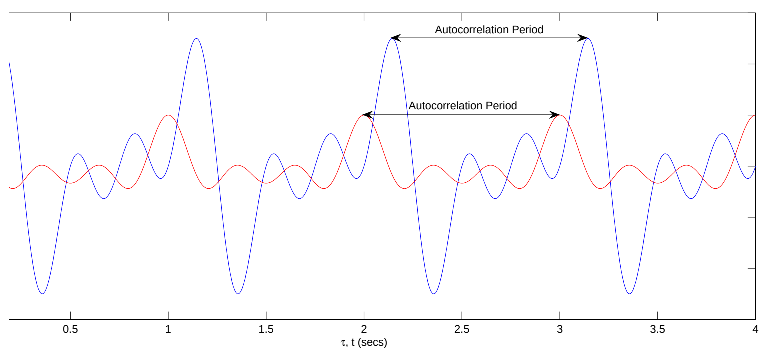


Figure 4: A signal (blue) composed of mixed cosines and the coefficients (red) from the autocorrelation values scaled to the range (1,-1). The largest peaks are periodic with the same frequency as the original signal. The lowest frequencies in the signal are marked as autocorrelation points for the signal (Above) and the coefficients (Below).

2.3 Wavelets

A.T. Cemgil outlines a number of other methods that can be applied to pitch estimation including FIR and IIR filter banks, which can be viewed as variations on the STFT (1). He also outlined wavelets as a method for signal processing.

Wavelet analysis has fairly diverse and disconnected origins. Morlet, one originator, developed what he called Gabor wavelets using STFT with windows of a constant number of oscillations, to look for a particular frequency in

[1]The complex conjugate is denoted as x^* and is the negation of the complex component of a complex number. In Cartesian form $ax + bi$ becomes $ax - bi$ and in polar form ae^{ib} becomes ae^{-ib}.

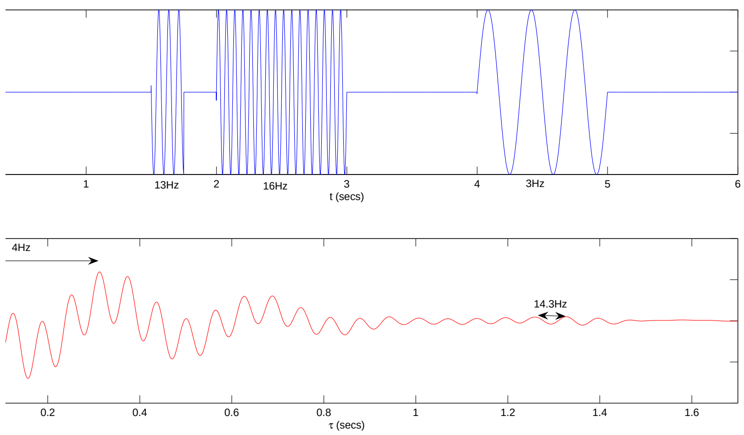


Figure 5: A graph of some of the initial autocorrelation R_x values for the mixed 3Hz and 16Hz signal, as in Figure 2, but with a 13Hz sine added on the interval 1.5-1.75s. With the original signal it was possible to find the 3Hz and the 16Hz peaks. However, when the 13Hz portion is introduced the major periodicity in the R_x values is 4Hz (not 3Hz) and a 16Hz period is present and no peaks corresponding to 13Hz are visible.

a signal. The window length is determined to be some constant number of periods at that frequency (5). This approach is very similar to what is now known as the Continuous Wavelet Transform (CWT):

$$c(a, b) = \int f(t)\Psi(at + b)dt \tag{3}$$

The value $c(a, b)$ is the CWT coefficient calculated by convolving the signal, $f(t)$, with a wavelet, Ψ, of scale a at offset b in time. The CWT often refers to a vector of all coefficients for a given scale. The continuous wavelet function must have an integral of zero; otherwise the scaling would introduce a bias on the imbalanced side of the function. Different wavelets at different scales can emphasize different properties in the signal (See Figure 7(a) for a specific example). A large amount of redundant information is encoded in the CWT because the coefficients do not change significantly over small changes in the offset or scale (5). Thus, information is usually extracted solely from the maxima of the CWT coefficients.

When processing large signals or multidimensional signals, calculating the redundant data can make the process computationally infeasible. The Fast Wavelet Transform (FWT), which was inspired by the pyramid algorithm (5), eliminates redundancy through orthogonality. A pair of functions are orthogonal if their inner product is zero. This implies that any information encoded by the first wavelet is not encoded by the second and vice-versa; therefore, no time is spent encoding redundant data. The FWT operates using a dyadic scale, progressing in powers of two. These FWT wavelets must satisfy:

$$\int_{-\infty}^{\infty} \Psi(2^i t)\Psi(2^{i+1}t)dt = 0, \forall i \geq 0 \tag{4}$$

The general idea behind the FWT algorithm is as follows:

$A_0 \leftarrow$ Signal
For $n = 1$ until done
  $D_n \leftarrow CWT(A_{n-1}, \Psi)$
  $A_n \leftarrow$ Downsample $A_{n-1} - D_n$ by 2

This procedure can continue until a desired level or until there is no additional information to extract. The depth of the algorithm is bounded by $n \leq \log_2(\text{samples})$ because of the downsampling. To perform noise reduction, it may only be necessary to compute A_1 and then discard D_1. The down-sampling reduces the size of the approximation at each level by two. To fully reconstruct the signal, each A_i is needed, along with the final D_n; All intermediate D_i may be discarded. Figure 6 shows a decomposition to $n = 5$.

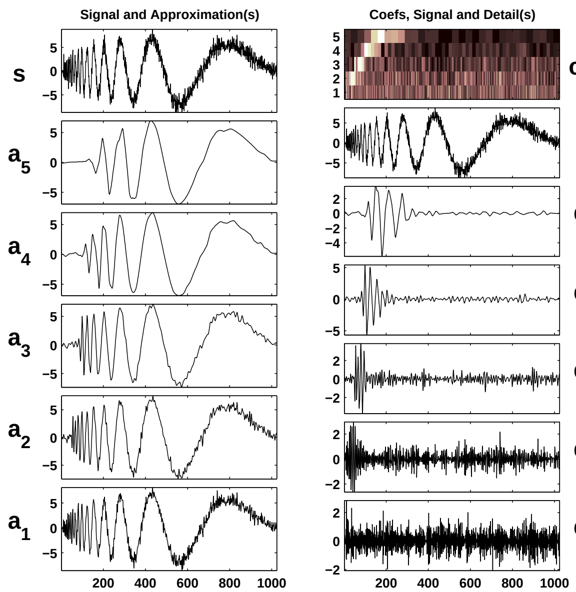
Since we have extracted the high frequency information and because of the Heisenberg Uncertainty Principle, no information is lost in the downsampling. The Heisenberg Uncertainty Principle puts a lower bound on the product $\Delta t \cdot \Delta f$, which limits the time resolution of lower frequencies (5). There is a more thorough explanation of this behavior in (1).

In practice, both the CWT and down-sampling are performed using special filters associated with the wavelet and the associated scaling function, as digital filter calculations are much faster than all of the convolutions required for a single scale of wavelet coefficients. The signal can be reconstructed using the lowest level approximation and the levels of detail above it (Figure 6).

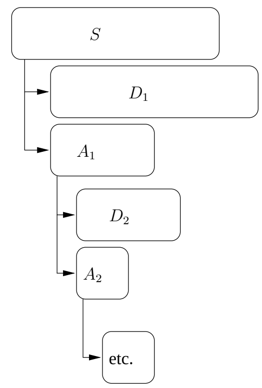
3 Applying Wavelets to Pitch Transcription

3.1 Foundation

A certain type of wavelet is useful for pitch detection with the CWT. Mallat proved that an analysis using a wavelet that is the first derivative of a smoothing function will exhibit maxima at points of change in the signal (1). Intuitively, this makes sense. A smoothing function, such as the gaussian (shown in Figure 7(a)), exhibits the property that, when convolved with a function, will preserve values closest to a single point in the domain while increasingly decreasing the values at further points in the domain. As shown in Figure 7(b), the first derivative of a smoothing function will be positive on the left side of the center and negative on the right. Thus, a CWT with a wavelet of this shape emphasizes zero crossings in signals (Figure 8). We can then use the maxima in



(a) An example decomposition from Matlab



(b) A diagram of the decomposition.

Figure 6: An example in Matlab of a noisy doppler signal decomposed using a symlet. The s at the top of the right column is the original signal. The decomposition of the signal is shown from the bottom upward. The s in the right column is the reconstructed signal, identical to the original, and *cfs* is a visualization of the coefficients, the scale on the vertical axis and the offset on the horizontal axis. The coefficients correspond to the contribution of the 2^i scales of wavelets which can be visualized in the detail d_i. Each a_i is the approximation of the signal without the d_j $\forall j \leq i$.

the coefficients of the CWT to identify zero-crossings in the original signal and calculate the corresponding frequencies. It is more useful to pick maxima in the CWT coefficients than zero-crossings in the original signal as different scales of wavelet emphasize different ranges of frequencies, and the original signal often contains more frequencies than just the fundamental frequency that we wish to extract.

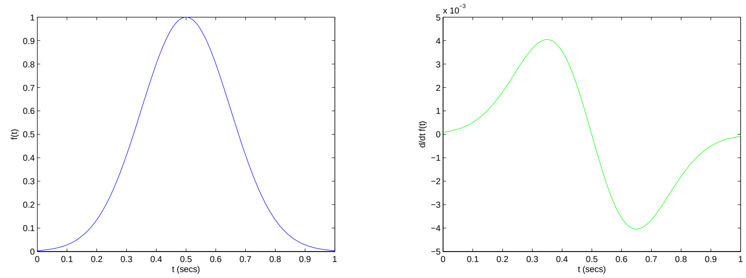
3.2 Estimating the Fundamental Frequency

Using these wavelet methods I was able to construct a monophonic pitch estimator. The general steps in the process are to:

1. Perform the CWT at a single scale, yielding coefficients for each offset.

2. Identify the maxima in the coefficients.

3. Estimate the frequency between each pair of maxima.



(a) A gaussian with $\mu = 0.5$ and $\sigma^2 = 0.15$

(b) The first derivative of the gaussian.

Figure 7: The gaussian is an example of a smoothing function whose first derivative can be used as a wavelet for pitch detection. The quadratic spline is another smoothing function that can be used (3).

4. Smooth the frequencies with a running average and clamp them to actual note frequencies.

5. Group the frequencies into notes with time and duration.

In principle, the analysis should be performed with a wavelet over all of the dyadic scales – because the different scales emphasize different frequencies – and we want to identify notes whose pitches span a wide variety of frequencies. Fitch and Shabana (3) suggest that evaluating over three adjacent scales was experimentally sufficient to estimate the pitch in the decaying section of a guitar-pluck. In the interest of simplicity and because the papers I read were opaque on the subject of combining the results from multiple scales, I used a single scale which was experimentally determined.

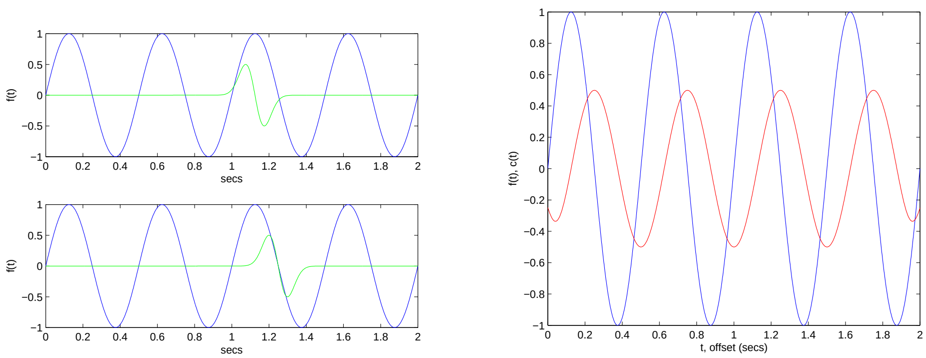
After using the first derivative of the gaussian to calculate the coefficients, the maxima are accepted if:

1. they lie above a *minimum energy bound* which was introduced to suppress the noise in 'silent' areas, and

2. if they are within a certain *frequency bound* (100Hz to 3000Hz).

The energy of a signal is the squared value of its samples – in this case I am referring to the sum of the squared value of the coefficients between zero-crossings.

Once the maxima in the coefficients are identified, the period between each adjacent pair is used to compute the frequency at that point in time. In some instances, when other harmonics or sounds are present, smaller maxima may appear in between the maxima of the fundamental frequency (6) (Figure 9). To compensate for this, the algorithm implements a crude *proportion bound*: adjacent maxima are only counted if they are:



(a) The wavelet (green) is superimposed on a 2Hz sine wave offset to align with a peak of the signal (Top), and to align with a zero-crossing of the signal (Bottom).

(b) The resulting CWT coefficients (red) scaled to the range (-0.5, 0.5).

Figure 8: A visual example of the CWT using the first derivative of a gaussian as a wavelet. When the wavelet is aligned with a peak in the signal, the positive and negative portions of the wavelet will cancel, whereas when it is aligned with a zero-crossing in the signal, all portions are positively maximized.

1. within certain proportions of each other or
2. the adjacent maxima are of monatomic ascending or descending magnitude.i

If those conditions are not met and the next maxima beyond is within the *proportion bound*, it is used to compute the frequency. In an instance when the two maxima are farther apart than the period of the minimum frequency, a rest frequency is assigned at the time of both maxima.

3.3 Estimating Pitch

The frequencies are then adjusted to the nearest value on the equal temperament harmonic scale. In this scale, developed by Bach, the notes C, C♯, D, D♯, E, F, . . . are half steps (multiples of $2^{\frac{1}{12}}$) apart, and the same note one octave above has twice the frequency (Petersen). After this adjustment, to extract note start time and duration information, individual time-pitch data-points that have the same pitch and are adjacent to each other are grouped together. In some cases, the pitch estimate in a note may deviate for a short period of time. This phenomenon is particularly common in the attack – i.e. the start – of a note as shown in Figure 10. To adjust for this, any note with a duration less than a specifiable parameter is merged with the harmonically closest adjacent

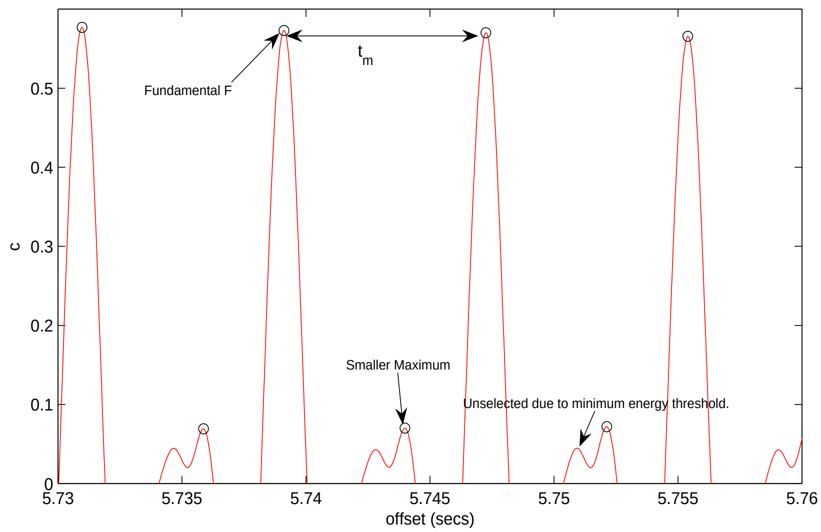


Figure 9: A small portion of the graph of the CWT coefficients for an third octave C guitar note. The maxima identified by the algorithm are indicated in the figure (black). There are three classes of maxima in the figure: The large maxima corresponding to the fundamental frequency, smaller maxima possibly related to overtones, and maxima that are not selected by the algorithm because they do not meet the minimum energy threshold. The smaller maxima do not meet the crude proportion threshold and the larger maxima are used. Each maxima corresponds to a zero-crossing and each pair corresponds to one-half period. A frequency of $\frac{1}{2 \cdot t_m}$ is assigned at the time of the left maximum. Frequencies are assigned to each maximum that satisfies the thresholds in this fashion.

note above the minimum length. Because the noise of the attack is at the front of the note and because the process is biased in the direction it operates, this is executed from back to front.

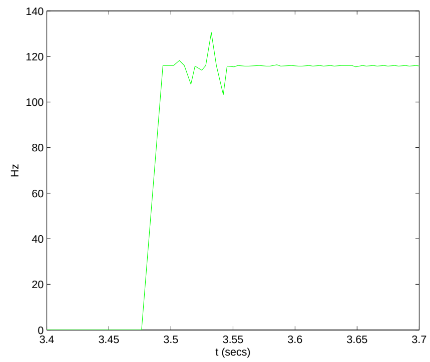
4 Results

I evaluated the algorithm using hand-labeled guitar data and a fairly simplistic comparison method. To ensure that the labels and estimates have the same time resolution, two arrays are created at a given sample rate, one from the labels and one from the estimated notes. The two arrays are then compared as follows: A sampled estimate is correct if it matches the labeled estimate. Incorrect estimates are divided into octave errors, two classes of rest error – false positive and false negative – and other note errors.

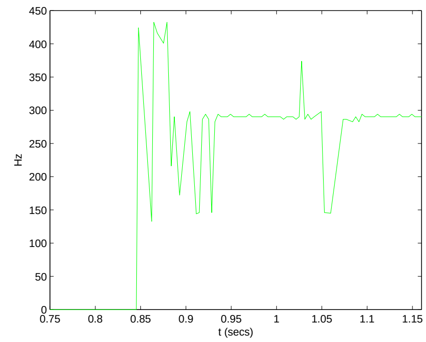
The sample data consists of single octave scales that rise and fall, and two octave scales that only rise. The C scales are played at two different positions on the guitar and six different speeds.

Name	Samples	%Correct	%Incorrect	%Inc Rest FP	%Inc Rest FN	%Inc Note	%Inc Octave
a3-a5	25300	97.64	2.36	15.38	53.68	0.00	30.94
b4-b6	25310	94.06	5.94	1.33	98.67	0.00	0.00
c4-c5_maj_1_1	8120	95.38	4.62	9.60	57.87	0.00	32.53
c4-c5_maj_1_2	8219	95.46	4.54	66.76	33.24	0.00	0.00
c4-c5_maj_2_1	4365	96.40	3.60	16.56	83.44	0.00	0.00
c4-c5_maj_2_2	4533	95.52	4.48	38.92	61.08	0.00	0.00
c4-c5_maj_3_1	3515	96.05	3.95	12.23	87.77	0.00	0.00
c4-c5_maj_3_2	3011	93.82	6.18	15.05	84.95	0.00	0.00
c4-c5_maj_4_1	2601	92.58	7.42	9.84	90.16	0.00	0.00
c4-c5_maj_4_2	2384	90.31	9.69	21.65	67.10	11.26	0.00
c4-c5_maj_5_1	1783	88.22	11.78	16.67	81.90	1.43	0.00
c4-c5_maj_5_2	1715	88.28	11.72	8.46	91.54	0.00	0.00
c4-c5_maj_6_1	1370	87.59	12.41	13.53	81.76	4.71	0.00
c4-c5_maj_6_2	1456	83.45	16.55	45.64	42.74	11.62	0.00
c5-c6_maj_1	8100	98.80	1.20	25.77	74.23	0.00	0.00
c5-c6_maj_2	4170	96.43	3.57	20.13	75.17	4.70	0.00
c5-c6_maj_3	2950	95.32	4.68	0.72	99.28	0.00	0.00
c5-c6_maj_4	2360	88.39	11.61	13.50	82.85	3.65	0.00
c5-c6_maj_5	1820	93.41	6.59	5.00	95.00	0.00	0.00
c5-c6_maj_6	1445	85.12	14.88	0.00	93.95	6.05	0.00
d4-d6	25365	98.14	1.86	0.00	100.00	0.00	0.00
e3-e5	25010	95.16	4.84	1.24	98.76	0.00	0.00
e5-e7	25260	85.34	14.66	0.05	99.95	0.00	0.00
g4-g6	25400	97.63	2.37	2.32	97.68	0.00	0.00

Table 1: Performance statistics on guitar data. %Correct and %Incorrect are measured with respect to the total number of samples in a comparison performed with a sampling frequency of 500Hz. The %Inc values are calculated relative to the number of incorrect samples. The sample names take the form *an-bm-maj_s_f* where *a* and *b* are notes, *n* and *m* are octaves, *s* is the speed (higher is faster), and *f* is the fret position.



(a) Guitar note, sampled at 44.1kHz



(b) Violin note, sampled at 22.05kHz

Figure 10: The estimated frequencies (green) after maxima picking – step 3 from 3.2 – at the attack of a guitar note (a) and a violin note (b).

4.1 Error Analysis

In general, the algorithm seems to perform well, as shown in Table 1. The octave errors in the two data-files (a3-a5 and c4-c5_maj_1_1) exhibiting them are related to the smaller maxima, potentially related to overtones, in the coefficients (Figure 11). In no instances is an entire note estimated as a different note; the only incidents occur at poorly separated note boundaries (Figure 12). False positives occur primarily at note boundaries. When evaluating the correctness of the pitch estimates relative to the labeled data, it is important to note that it is unclear whether the hand-labeled "truth" corresponds to precise boundaries in the audio (Figure 13).

Raising the minimum energy threshold can assist in shortening the amount of the decay that is included in the notes' duration. However, this has a strong effect of the detection of high frequencies, especially since only a single wavelet is being used, which tends to be most selective in a certain range of frequencies. At frequencies on the outside of this range, the coefficients tend to be much smaller and are therefore more likely to be cut off by the minimum energy threshold (Figure 14). This is exacerbated by the rapid decay of high pitched notes on the guitar. High-frequency errors account for a majority of false-negative rest errors and there is a strong sensitive dependence amongst the ability to detect a broad range of frequencies with a single wavelet and the need to find rests between adjacent notes using only the minimum-energy threshold.

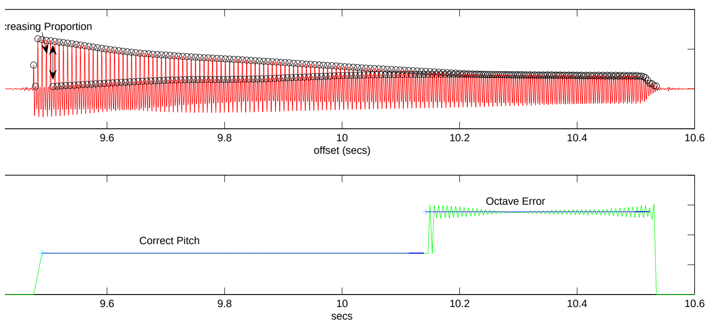
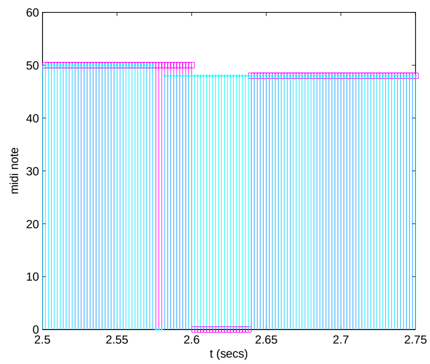


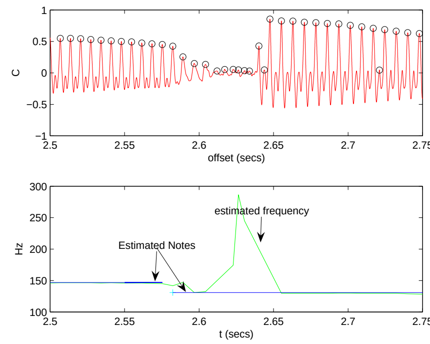
Figure 11: An example of an octave error from the *a3-a5* scale sample. When the maxima from the overtones begin to meet the *proportionality bound* – as illustrated in Figure 9 – relative to the maxima corresponding to the fundamental frequency (Top), they are no longer ignored by the algorithm and the estimated frequency approximately doubles (Bottom), resulting in octave error. In situations where the octave error is shorter than the minimum note threshold, the error will be merged into the earlier portion of the note at the appropriate pitch. Octave errors occur primarily at the end of notes because the amplitude of the fundamental frequency seems to decay faster than the overtones.

4.2 Conclusions & Future Work

Overall I am satisfied with the results of the algorithm as evidenced on this data set. While there was a small amount of parameter twiddling on the thresholds, it has generally done well. Yet, there is definitely room for improvement, particularly in more dynamic situations with rapid note changes. Currently, only the positive maxima are used because the inclusion of the negative maxima introduce additional variance in the estimated frequency making it difficult to identify stable notes. I am uncertain why the negative maxima seem to be less stable than the positive maxima. Investigating the cause of this, along with the presence of the lower maxima that induce the octave errors, could be illuminating. Spreading the analysis across multiple scales of wavelet analysis, either as an ensemble classifier or otherwise, should help resolve the issues with high-pitched notes. Such a technique should be particularly effective in correcting octave errors and possibly reduce the need for identifying low maxima. There is potential for improvement in the note grouping algorithm; possibly some type of predictive look ahead could help relieve the octave error issues. Trying to investigate the extra frequencies within the attack of the note, as well as the frequencies of the harmonics, could help to smooth the pitch estimation and grouping issues. This could also bring some insight into polyphonic pitch transcription.



(a) Algorithmically-estimated (cyan •) and hand-labeled (magenta □) notes at a note error in *c4-c5_maj_6_2*.

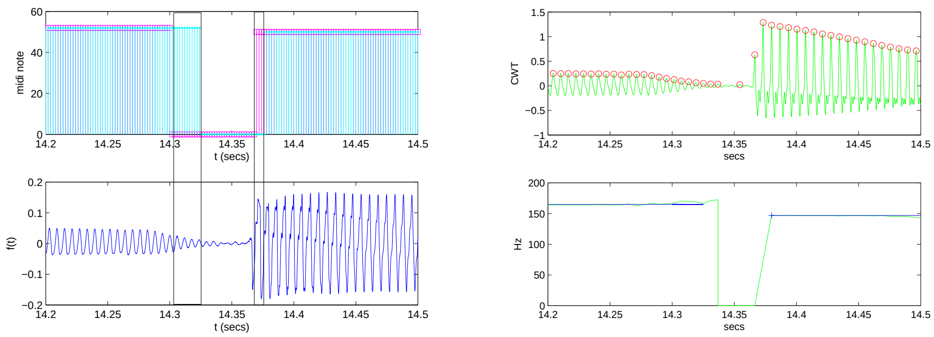


(b) The maxima (Top) and calculated frequency with estimated notes (Bottom) for the note error in 12(a). The octave error in the calculated frequency is eliminated through the grouping method.

Figure 12: Note errors primarily occur between two note boundaries than are not well separated temporally. Usually this occurs when the attack of the second note overlaps with the decay of the first. The decay as well as other noise in the time between the notes is not excluded by the minimum energy filter, therefore no rest is detected. Recall that the grouping algorithm is biased toward the rear, so that the intermediate noise and noisy sections at the end of the first note are merged in to the second note. Relative to the algorithm's pitch estimate, the noisy sections from the first note that are merged with the second note are classified as note errors; and the intermediate section is, itself, classified as a false positive rest.

5 Acknowledgements

I would like to thank Professor Belinda Thom for her encouragement, assistance, and for letting me research a project completely devoid of the statistical methods we've been studying in class. I would also like to thank Joseph Walker for the use of his labeled guitar data.



(a) Algorithmically-estimated (cyan •)and hand-labeled notes (magenta □) (Top) and the actual wave data (Bottom) for *c4-c5_maj_1_2*. Boxes indicate locations where the hand-labeling is questionable.

(b) The maxima (Top) and calculated frequency with estimated notes (Bottom).

Figure 13: Determining the precise boundaries for a note can be difficult, as one must determine where in the attack and in the decay a note begins and ends. Discrepancies in the accurate labeling of note boundaries, by hand or otherwise, were generally linked to rest errors. In general, note boundary issues are categorized as segmentation errors.

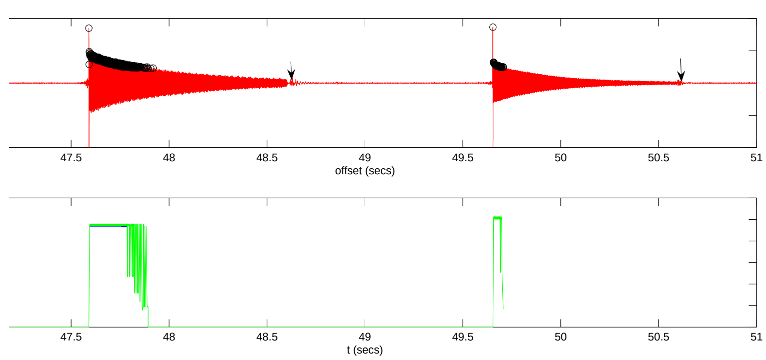


Figure 14: An example from *b4-b6* where the high-frequency CWT coefficients (Top) are ignored because of the minimum-energy threshold. Near the end of the notes (indicated with arrow) the coefficients are very similar in amplitude to the silent portions of the signal. The initial frequency estimates (Bottom) illustrate which how much of the note is missed.

References

[1] Cemgil, A. T. (1995). Automated music transcription.

[2] Evangelista, G. (2001). Flexible wavlets for music signal processing. *Journal of New Music Research*, 30(1):13–22.

[3] Fitch, J. and Shabana, W. (1999). A wavelet-based pitch detector for musical signals. In Tro, J. and Larsson, M., editors, *Proceedings of DAFx99*, pages 101–104. Department of Telecommunications, Acoustics Group, Norwegian University of Science and Technology. `http://www.tele.ntnu.no/akustikk/meetings/DAFx99/fitch.pdf`.

[4] Frazier, M. W. (1999). *An Introduction to Wavelets Through Linear Algebra*. Springer.

[5] Hubbard, B. B. (1998). *The World According to Wavelets*. A K Peters, second edition.

[6] Jehan, T. (1997). Music signal parameter estimation. `http://web.media.mit.edu/~tristan/Papers/NVMAT_Tristan.pdf`.

[Jones] Jones, G. `http://www.kingston.ac.uk/dirc/`.

[Petersen] Petersen, M. Mathematical harmonies. `http://amath.colorado.edu/outreach/demos/music/`.