Harmonic Spectral Fit: Onset detection of slow-building attack transients

Hannah Robertson

Final Project for MUMT 501
Submitted: 9 December 2011

Onset detection is an important aspect of music information retrieval: in order to transcribe or follow note events you must first be able to identify where those events occur in music. Various forms of time-to-frequency-domain analysis show what frequencies - and therefore what notes - occur in a section of music, but in order to study and analyze a score as a whole it is extremely useful to also know the onset of each event. There are several different onset detection methods, which deal in various combinations of spectral frequency and phase, compared between successive frames of a recording. The harmonic spectral fit onset detection method of [3] was designed to be effective for both slow transient onsets such as those of wind instruments and onset detection of extremely short notes, such as ornaments. In this project, this harmonic spectral fit algorithm is presented, reproduced, and discussed.

1 Background and theory

1.1 Introduction to onset detection

Onset detection is the detection of musical note events within a sound recording. It is extremely important to any music information retrieval (MIR) task that involves timing; beat-tracking, score-following, and automated music transcription are three such tasks. As note events themselves differ based on the instrument that created them, with vastly different temporal and spectral identities depending on how the sound was produced, the algorithms for detecting onsets are often tailored to detect *specific types* of onset. For example, beat-tracking systems designed to follow drum beats can focus on identifying short, high-energy events within a frequency spectrum, while a system designed to automatically transcribe, or notate, a song played by a clarinet could focus on harmonic and spectrally dynamic events.

Some of the considerations of various onset detection algorithms are the harmonic content of the onsets, the total energy of the onset compared to the energy of a sustained note, the energy spread of the onset, the change in energy over the length of a note, and the change in phase of the spectral components. Onset detection algorithms compare different spectral aspects of successive frames of sound to determine when the note

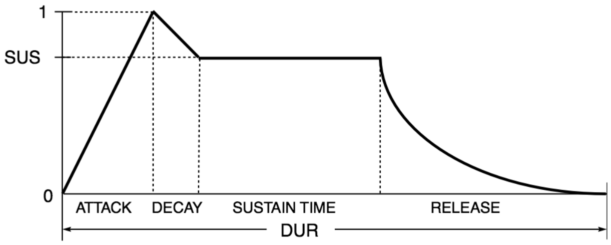

Figure 1: Diagram of attack-decay-sustain-release [1].

starts. There are algorithms based on auditory modelling, knowledge modelling, data representations, detection of periodicity in the time or frequency domain, and energy variations, as summarized in [2] and [4]. Unsurpsingly, best onset detection method for any given task depends on the type of onset, or attack transient, that is being detected!

1.2 Attack transients

Musical note events are often described in terms of their attack, decay, sustain, and release (ADSR) components, as depicted in Figure 1. Attack transients are the initial part of a note's sound, the time when the sound is building. By measuring and comparing the energy and spectrum properties of a sound to the same properties a moment earlier, it is possible to determine whether a note event has occurred. It is clear from Figures 2, 3, and 4 that the attack transients of different instruments contain very different temporal and spectral content, both in terms of the components present and the relative energy of those components. Additionally, the strength of the components changes over time. It can be seen that the piano attack initially involves a wide range of frequencies, including many at the higher end of the spectrum, but then drops off quickly. The flute attack, on the other hand, starts with low frequencies and builds up to higher ones, although never having as much energy at higher frequencies as the piano. The fiddle's attack transient is similar to the flute's, in that it builds more slowly than the piano and never fills in in the upper frequencies. In general, woodwind and bowed string instruments have slower attack transients than brass and struck instruments (such as piano and chimes).

1.3 The state of onset detection algorithms

An onset detection algorithm must be inclusive of all types of attack transients present in the music it is processing. This can prove problematic because the attacks of different instrument onsets are best measured in different ways. Refining an algorithm for one type of onset often comes at the expense of resolution for another. For example, looking for the occurrence of high frequencies gives excellent onset detection for piano note events, but less ideal onset detection for fiddle notes. This means that onset detection applications often run multiple detection algorithms, and then combine and refine the various detected onsets to determine most probable true onsets.

In the past, various algorithms have focused on energy subbands and phase changes.

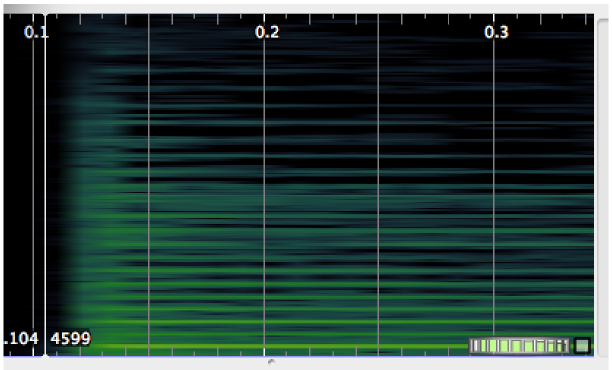

Figure 2: Spectrogram of piano onset (time labeled across the top in seconds).

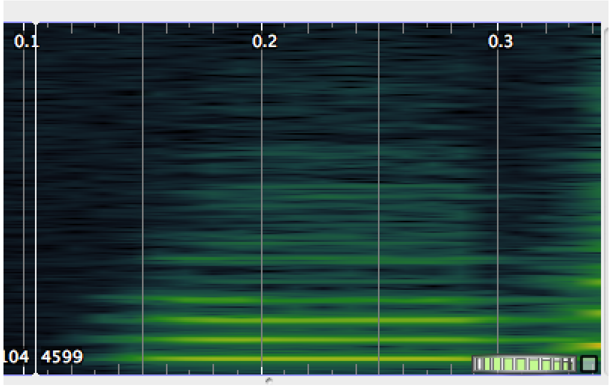

Figure 3: Spectrogram of flute onset (time labeled across the top in seconds).

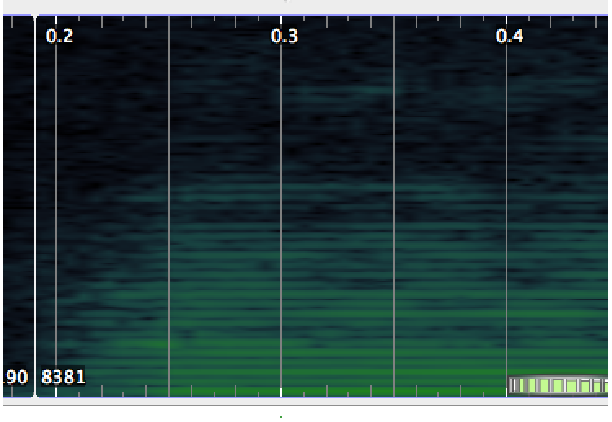

Figure 4: Spectrogram of fiddle onset (time labeled across the top in seconds).

Because these methods reliy on identifying song frames with higher frequency components than the spectra of previous frames, they aren't particularly good at quickly detecting onsets produced by instruments with slow attack transients; there is a slight lag in detection time. The lag isn't a big deal if the precision of onset detection doesn't need to be too high, for example if the ratio of a note's attack to sustain portions is low and the ratio of attack to smallest note length is also low, such as for a slower melody where the length of the smallest note is much longer than the attack transient time. As long as the detection isn't needed in real-time, e.g. a transcription application rather than a real-time score-following one, this lag could be systematically adjusted for.

On the other hand, lag is a problem in polyphonic detection applications that mix instruments with both slow and fast attacks. For example, a transcription of a piano and flute duet, if both instruments are playing very fast, ornamented notes, might show the flute consistently playing a beat or two behind the piano. Even for a monophonic recording, a note event such as a trill, bend, cut, or mordent that is shorter than the full attack transient will be entirely missed by an energy sub-band algorithm. A piano trilling? Lots of spectral energy in all subbands, so no problem. A flute ornament? If it happens fast enough, *big* problem.

Enter harmonic spectral fit algorithms.

1.4 Harmonic spectral fit algorithms

One way to detect how the component frequencies change over time is to measure how well a sound fits a set of known possible frequencies: its spectral fit. For example, if at time A a frequency 550 Hz is present, and at time B a frequency 724 Hz is present, it can be reasonably assumed that a note event happened between time A and time B. Musical notes, in Western music at least, are discretized in such a way that only a finite selection of fundamental frequencies will be played in any given piece. By looking at those frequencies it is possible to determine onsets. Just determining presence or lack of presence can be too broad, however. Measuring the energy of each known note frequency, can account for spectral leakage. This energy measure can be done by running the input sound through filterbank of nulling filters, each set to the specific frequency of a known note, and then comparing the energy of the filtered sound to the original sound for each note. In 2005 this method was implemented by Kelleher for fiddle onset detection [5].

This idea of spectral fit can be taken a step beyond detection of known fundamental frequencies, if it is known that a harmonic frequency spectra is present in the input sound. Rather than looking at only the fundamental frequencies of each note, measuring the energy in all harmonic components of the note gives an even more pronounced measure of how a sound changes over time, and therefore whether an onset has occurred. This is especially useful because the instrument families for which spectral fit is most useful, i.e. woodwinds and bowed string instruments with slow attack transients and reduced high-frequency energy components, happen to be harmonic: the spectrum of a note played at fundamental frequency A will have significant spectral components at frequencies $2 * A, 3 * A, 4 * A$, and so on, and little to no response at other frequencies depending on the inharmonicity of the instrument. This measure of harmonic spectral fit, involving a

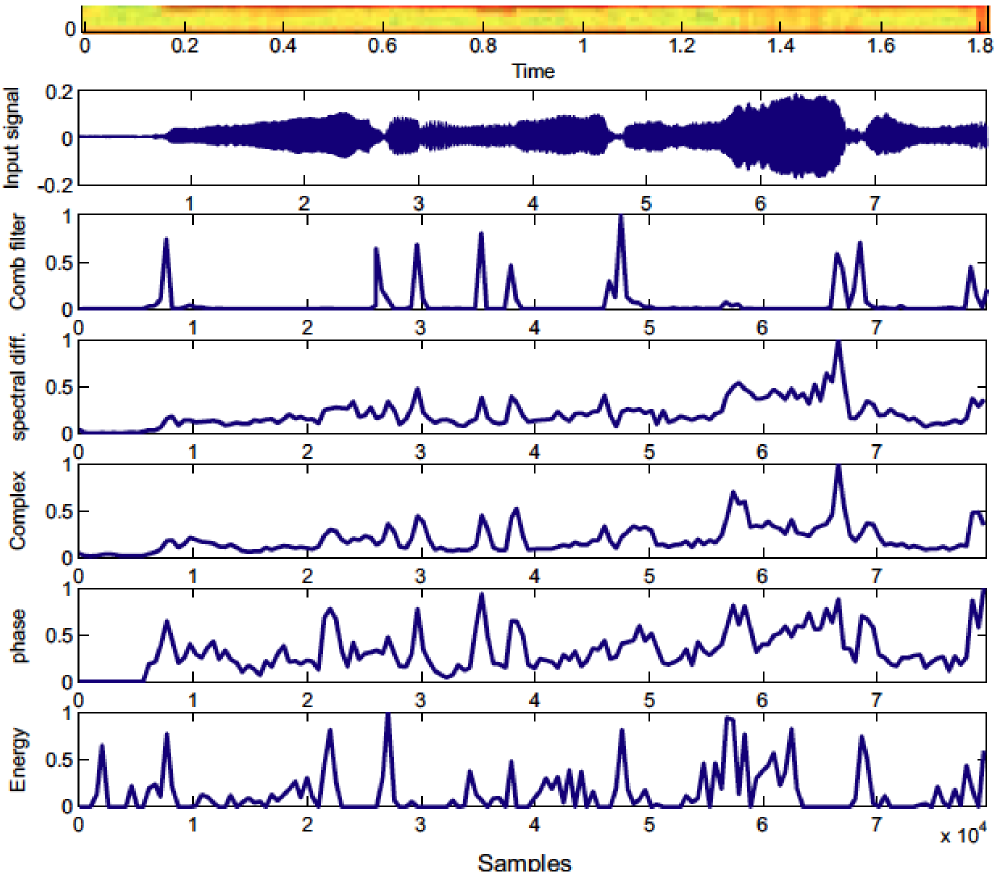

Figure 5: Different onset detection methods, as demonstrated in [3, Figure 7].

filterbank of comb frequencies, was proposed by Gainza, Coyle, and Lawlor in 2005 [3] and utilized by Gainza and Coyle in an automated ornamentation detection system in 2007 [2]. Their algorithm and a re-implementation are described in the following sections of this paper.

This harmonic spectral fit onset detection method is most affective for melodic instruments; while it will also detect the onset of non-melodic instruments such as un-tuned drums or cymbals, to use it on such non-melodic instruments is overkill. Simply comparing the total energy of consecutive frames will give enough information about whether a drum has been hit or not.

As can be seen in Figure 5 [3], onset detection using spectral fit is at least as good as the rest of previous algorithms, and better than some, as indicated by the very clean and correctly identified onsets in the first plot under the input signal (second plot from the top).

2 Harmonic spectral fit

In the following sections, the filterbank of comb filters is described algorithmically and then implemented in MATLAB.

2.1 The filters and algorithm

This section describes, step-by-step, how this onset detection function filters a signal x to get onset detection information.

First, the signal is sent through a bank of i comb filters, each with delay D_i equal to to the length in samples L of a known note fundamental frequency f_s,

$$L = f_s/f_0.$$

For example, for frequency $440Hz$ and sampling frequency $44.1kHz$, the delay $D_i = 44,100/440 =$ slightly over 100 samples.

The equation for the each comb filter is

$$y_i[n] = x[n] + g * x[n - D_i] \tag{1}$$

with gain $g = 1$. This filter takes the input signal and delays it by D_i samples; because the gain g is positive, the delayed signal $x[n - D_i]$ is added to the original signal. If x is periodic with frequency D_i, the output $y[n]$ has twice the frequency component D_i of $x[n]$. If x is periodic with frequency $D_i/2$, negative interference causes a cancelation of the D_i frequency component from the original signal.

Much like for the STFT, each filtered signal $y_i[n]$ is windowed and hopped, with a Hann window

$$A(x) = cos^2\left(\frac{\pi x}{2a}\right) = \frac{1}{2}\left[1 + cos\left(\frac{\pi x}{a}\right)\right] \tag{2}$$

of length 1024 and a hop size between windows of 1/2 the window length, 512 samples. These values were the values used in [3].

For each frame m of the filtered signal $y_i[n]$, the total energy is calculated:

$$E_i(m) = \sum_{all\ n} (y[n])^2 = \sum_{all\ n} (x[n] - x[n + D_i])^2 \tag{3}$$

Keep in mind that "for all n" means all n in frame m, not all samples n in the original signal.

This frame energy needs to be normalized. To do this, the frame energy is divided by the energy of a maximally filtered frame, which is to say y_{max} when x is purely harmonic at the comb delay D_i, with no other frequency components. This occurs when $x[n] = x[n + D_i]$, such that

$$y_{max} = x[n] + x[n + D] = 2x[n]$$

and therefore

$$E_{i,max}(m) = \sum_{all\ n} (y[n])^2 = \sum_{all\ n} (2x[n])^2 = 4\sum_{all\ n} x^2[n]. \tag{4}$$

This means that the normalized energy for frame m is

$$E_{i,norm}(m) = \frac{E_i(m)}{E_{i,max}(m)} = \frac{\sum_{all\ n}(x[n] - x[n + D_i])^2}{4\sum_{all\ n} x^2[n]}. \tag{5}$$

This normalized energy value, $E_{i,norm}$, equals 1 when the comb filter maximally filters the signal, and 0 when there is no fit between the input signal and the filter. Because spectral fit is defined in [3] as being 0 at maximum match and 1 and no match, the measure of spectral fit $E'_i(m)$ is defined as

$$E'_i(m) = abs(E_i(m) - 1). \tag{6}$$

To determine the onset detection function for each filter i in the filter bank, ODF_i, from the measures of spectral fit, take the square of the first order differences between consecutive frames:

$$ODF_i(m) = \left[E'_i(m) - E'_i(m-1)\right]. \tag{7}$$

ODF$_i$ is a measure of harmonicity changes for note i between consecutive frames: between two frames with complete spectral fit, ODF = (0 - 0) = 0, so no onset occurred even though the note itself was in the process of being played; between two frames with no spectral fit, ODF = (1 - 1) = 0, so no onset occurred on the note was not being played; between two frames with differing spectral fit, e.g. ODF = .9 - .2 = .7 or ODF = .2 - .9 = -.7, which indicates that the spectral fit increased so the note was started (ODF positive) or decreased, so the note was stopped (ODF negative).

All these individual onset detections are then summed to give an overall onset detection function that covers the frequency range of the individual comb filters:

$$ODF(m) = \sum_{all\ i} ODF_i^2(m). \tag{8}$$

By squaring each individual ODF, information is lost about whether a specific note was started or stopped, but still indicates that there was *some* event. Keep in mind that an offset in one filter is often linked to an onset in another, for example going from note A to B. If ODF_A and ODF_B aren't squared, and if they contain the same energy, then $ODF_A = -ODF_B$ and summing them will yield 0, or no onset event, which we know to be false! By squaring each before summing, we get $ODF_{total} = 2 * ODF_A^2$, which will be a positive number between 0 and $I * .99^2$, where I is the number of filters. Because onset detection functions are interested in note (or rest) events in general, just knowing that there was an onset is sufficient. Even so, it is good to keep the extra note-on/note-off information from the ODF_is in mind for possible future tasks.

2.2 Implementation

The above equations were implemented in MATLAB. The code follows the algorithm presented above almost exactly, so to step describe the algorithm again would be unnecessarily repetitive! The MATLAB file has been turned in with this paper and is well-commented in regards to the step-by-step procedure of the algorithm.

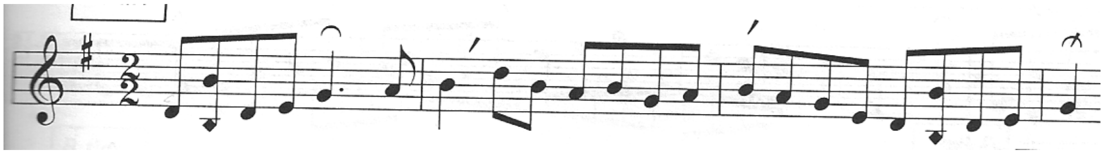

Figure 6: The first phrase of the traditional tune "Christmas Eve," as played by Grey Larsen[6].

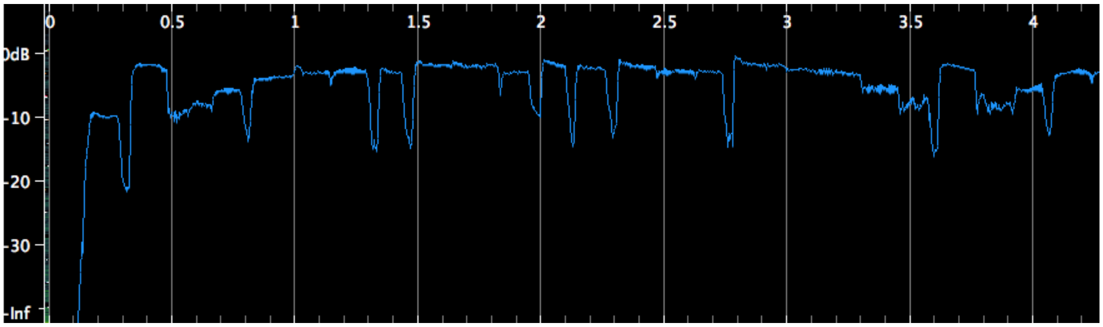

Figure 7: The amplitude envelope of the first phrase of the traditional tune "Christmas Eve," as played by Grey Larsen[6].

The sound file used was the traditional tune, "Christmas Eve," as played by Grey Larsen [6]. Figure 6 shows the notation of the tune; Figure 7 shows the amplitude envelope in the time dimension of the full notated phrase, and Figure 8 shows a spectrogram covering the same time. The sound file itself is included in the project's MATLAB folder.

To determine what frequency bins to use in the filter bank, an FFT of the original sound file was taken and frequency bins of known whistle notes were chosen. The results can be seen in the plots of the following section.

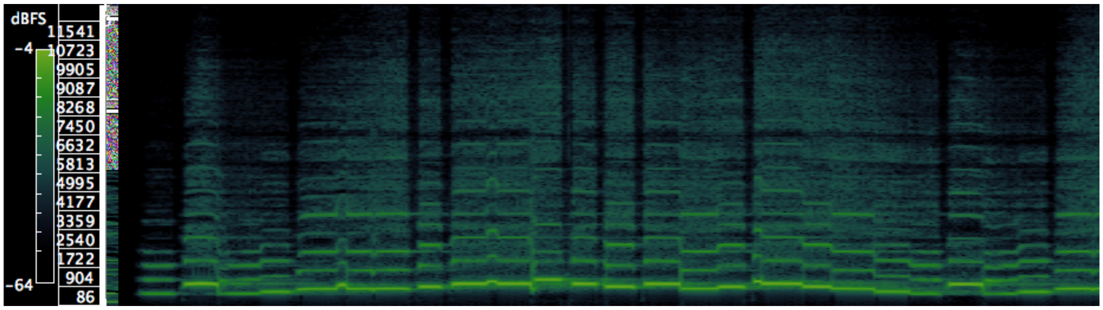

Figure 8: A spectrogram of the first phrase of the traditional tune "Christmas Eve," as played by Grey Larsen[6].

3 Results and discussion

3.1 Results

As can be seen in Figure 9, the implementation of a spectral fit detection method did a good job of finding possible onset events for each present note individually, but as can be seen in Figure 10, the total onset detection algorithm, consisting of the summed onset detection functions of each individual known note (the result of each comb filter in the filter bank) are less clear as to the individual onsets; plotting it on a log scale helps show that there are multiple peaks, but some form of weighting or thresholding function is will need to be included in the future. A thresholding algorithm was proposed in [2] and will be tried in this implementation in the future.

There is also something strange happening just before frame 75. While it makes sense for an event to have been detected by two different comb filters, as a note onset for note A is often coupled with a note ofset for note B as the player switches from B to A, this event has been detected by *all* the different notes, and since the notes aren't all harmonic multiples of one another this indicates that there might be some form of spectral leakage due to either windowing or poor choice of comb frequency delays. More investigation is needed to determine exactly what is happening, and how to fix it.

3.2 Discussion

This algorithm was not difficult to implement in MATLAB, save a few issues due to learning to work with filters in general. Now that the algorithm is up and running, it would be interesting to look at how different windows affected the effectiveness of this filter. [3] and [2] used a Hann window of size 1024 and hop size 512 when looking at successive frames of each song, which is why those sizes were implemented in this project, but now that it is apparent that such sizes yield *some* result, it would be interesting to see how different windows could fine-tune that result. Using windows with a wider main lobe could help account for slight variations in fundamental frequency and slight inharmonicty of the instruments played. This would be especially important in Irish or other folk music, where the tuning of instruments (at least to one another!) is notoriously lacking; if each filter acted on a wider range of frequencies around each known note frequency it would be possible to use this harmonic spectral fit detection method on polyphonic music where there were two or more of the same type of instrument, to determine overall note event occurrences with a minimum of computing power.

Alternatively, having a narrower lobe could allow for the separation of two notes that are supposed to be at the same fundamental frequency but are in reality slightly off - for example, *two* whistles playing the same note at the same time. If the resolution of each note was separable, it would be possible to create a second filterbank tuned to these slightly different fundamental frequencies, and determine note onsets for each instrument independently.

Additionally, this idea of onset detection using harmonic spectral fit could be accomplished using pass-band rather than comb filters centered at each of the known note fundamental frequencies, and encompassing an acceptable range of intonation deviation.

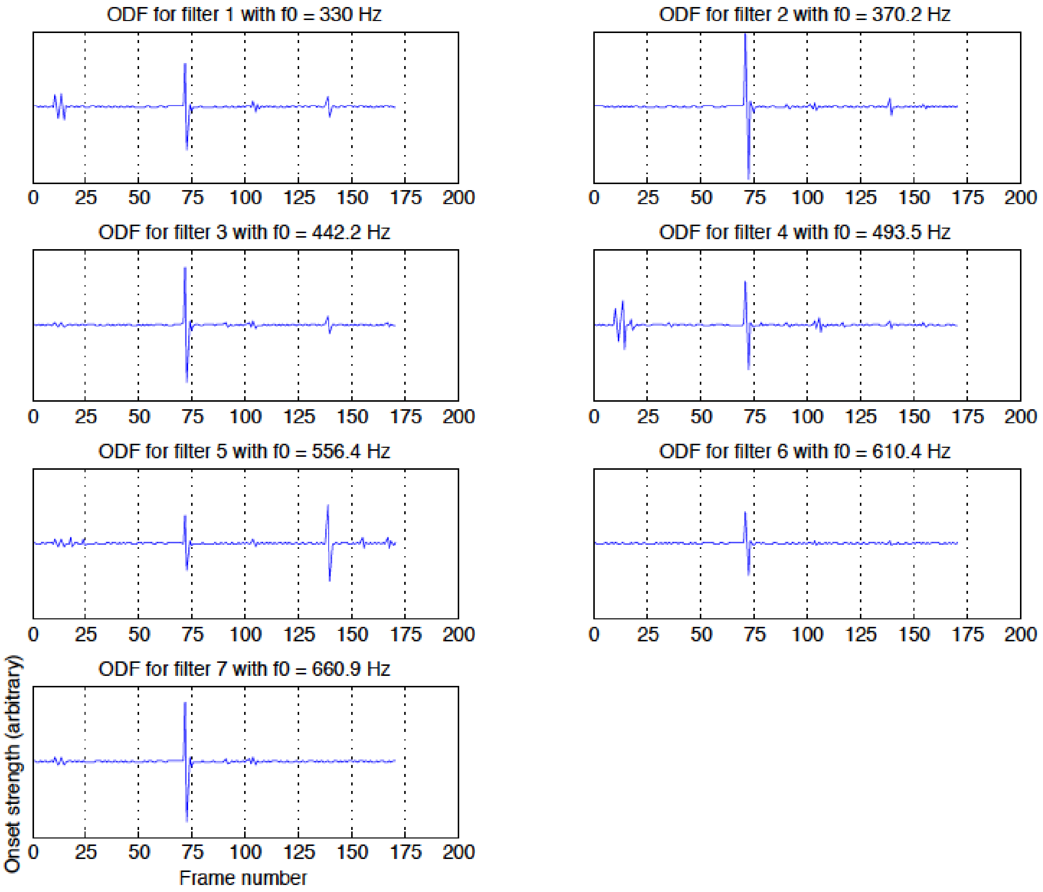

Figure 9: ODF for each filter i, as implemented in MATLAB.

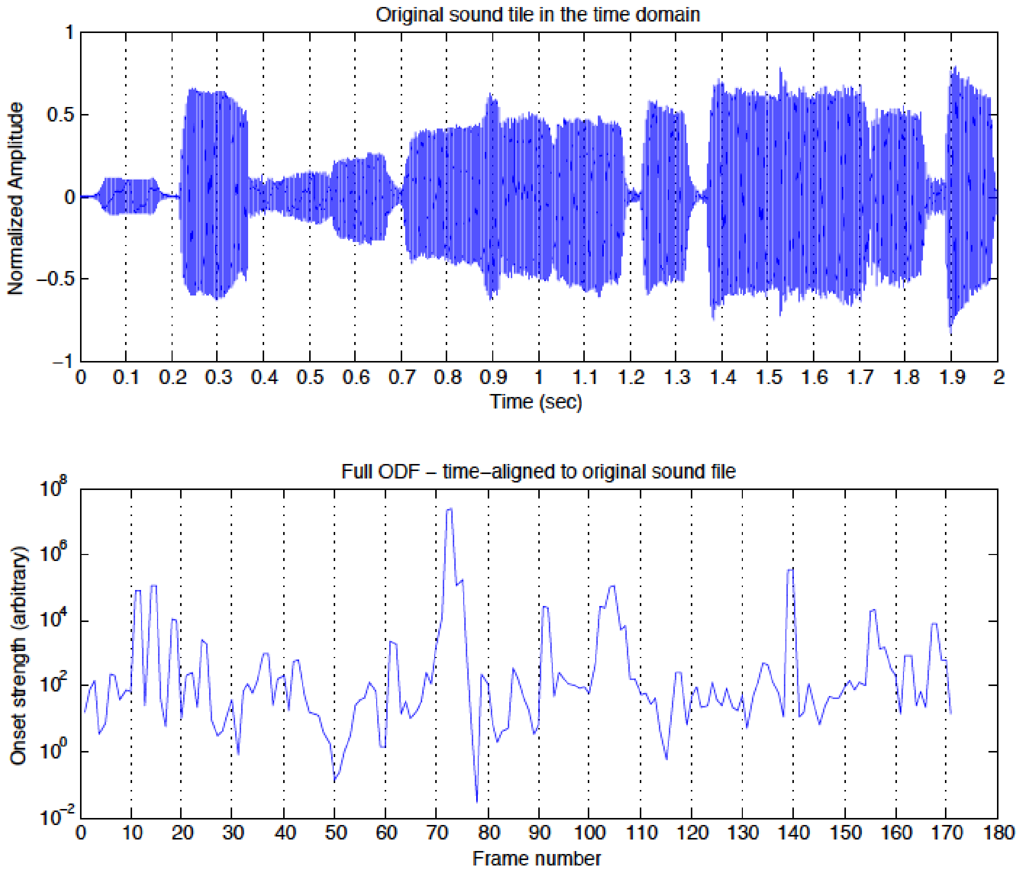

Figure 10: Full ODF, as implemented in MATLAB.

This would help improve detection for instruments with fundamental frequency ranges rather than discrete points - for example, fiddles rather than whistles. Whistles can produce a range of frequencies around a specific intended note by altering their air production, but still have a single choice of finger position on the instrument hole, covering or not covering, that automatically limits the sounded frequencies for that note. Fiddle players intending that same pre-determined note, however, might not put their finger on the string at the exact right spot. By filtering for a range of frequencies about the fundamental, as with a harmonic passband filter centered at the fundamental, this deviation could be accounted for.

In general, this algorithm lived up to its promise of being a successful way to get additional information about note onsets not provided by previous onset detection methods, and would be useful in most situations where a slow attack transient instrument is present. An interesting next step would be to compare its accuracy to the three sets of algorithms that were submitted to the 2011 Music Information Retrieval Evaluation eXchange (MIREX) competition in the Audio Onset Detection category this year. The only one currently published focuses on transient peaks [7], which it claims covers the pertinent harmonic information, at least in general. It will be interesting to see if it performs as well as the harmonic spectral fit algorithm presented here on short notes with slow attack-transients.

References

[1] James Beauchamp. Music 4c. *ems.music.uiuc.edu.beucham*, 1996.

[2] M. Gainza and E. Coyle. Automating ornamentation transcription. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–69 – I–72, April 2007.

[3] M. Gainza, E. Coyle, and B. Lawlor. *Onset detection using comb filters.* October 2005.

[4] Aileen Kelleher. Onset and ornament detection and music transcription for monophonic traditional irish music. *Dublin Institue of Technology Engineering MA Thesis*, Jan 2005.

[5] Derry; Coyle Eugene; Lawlor Bob; Gainza Mikel Kelleher, Aileen; Fitzgerald. Onset detection, music transcription and ornament detection for the traditional irish fiddle. In *Audio Engineering Society Convention 118*, 5 2005.

[6] Grey Larsen. *The Essential Guide to Irish Flute and Tin Whistle.* Mel Bay Publications, 2003.

[7] A. Robel. Onset detection by means of transient peak classification. *ISMIR*, 2011.