Using Audio Onset Detection Algorithms

1st Diana Siwiak
Victoria University of Wellington
Wellington, New Zealand

2nd Dale A. Carnegie
Victoria University of Wellington
Wellington, New Zealand

3rd Jim Murphy
Victoria University of Wellington
Wellington, New Zealand

***Abstract*—This research implements existing audio onset detection algorithms to analyze flute audio signals. The motivation for this research is to determine which techniques work better for real-time analysis. By methodically analyzing several well-known, pre-existing onset detection algorithms using a solo flute audio signal, exemplary audio features and analysis techniques will be determined. Flute audio signals are unique, in that they mimic pure sinusoidal tendencies more so than other woodwind audio signals. The analysis of these algorithms will contribute to the field of research in flute note onset detection.**

***Index Terms*—audio signal processing, music information retrieval, analysis, automatic algorithm, flute signal**

I. Introduction

Since the genesis of music information retrieval (MIR) as a research field, many researchers have built algorithmic tools for performing audio signal processing and analysis to study various aspects of musical instruments, music compositions, and performance attributes [1]–[4]. There are active international research groups and communities, including the International Society for Music Information Retrieval (ISMIR)[1] and the Music Information Retrieval Evaluation eXchange (MIREX)[2], whose goal it is to design, refine, and test various algorithms and tools for music information retrieval.

There are several levels to Music Information Retrieval:

- High-level representations: style, musical expression.
- Mid-level representations: melody, key and chord, note/event, beat per minute, rhythm.
- Low-level representations: Mel-frequency cepstral coefficients, complex domain, Fast Fourier Transform (FFT).

Low-level representations, or descriptors, are the measurable properties of the mid-level and high-level representations that are extracted from a music signal. They contain information relevant for pattern recognition, such as beat tracking. A contribution of this research is revealing ways to present musical expression as a quantifiable multi-dimensional space of feature vectors by using timing and rhythm as the basis.

II. Flute Signal Processing

Figure 1 shows the musical signals of a snare hit (in pink) and of a flute note (in blue) represented as both a waveform and an amplitude envelope in the time domain. Temporarily setting aside factors such as the frequency content, spectral content, and the amplitude, the visual difference between the two musical signals shows a steep, impulsive attack of the snare hit compared to a gentle, gradual attack of the flute note. The distinctions between the duration of the two attacks (even in those first few milliseconds) and shapes of the slopes (linear versus non-linear) are important factors to consider when determining a musical event. The percussive nature of the snare hit, where there is a fast change in amplitude over time, allows for a much clearer detection of note onset than the non-percussive, slow ramp of the flute note. The amplitude envelope of a snare hit can be described as attack-release (AR), whereas the amplitude envelope of a flute note is can be described as attack-decay-sustain-release (ADSR).

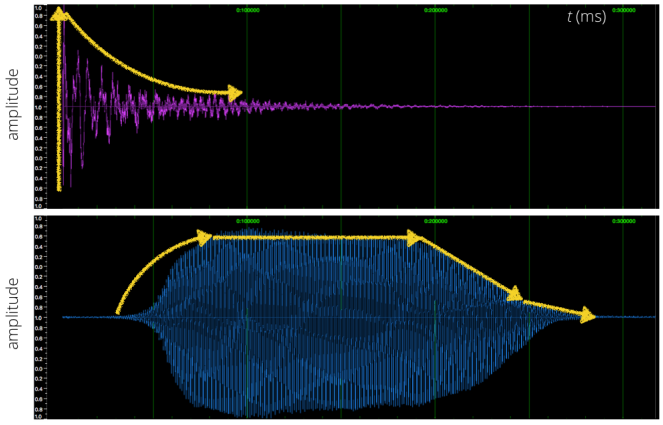

Fig. 1: Waveforms and exponential approximations of the amplitude envelopes for snare (top) and flute (bottom)

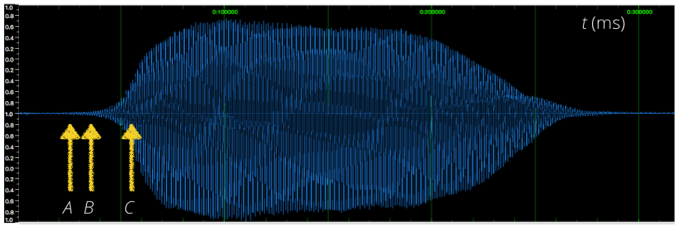

Fig. 2: Physical Onset (A), Perceptual Onset (B), and Perceptual Attack Time (C) of a flute note

Of all the physical instruments (including voice), flute is closest to a purely sinusoidal and harmonic signal. While a single flute note might have sinusoidal tendencies, flute music is complex, unique, and often difficult to extrapolate generalizable features, especially among a range of flutists. Soft, or

[1]http://www.ismir.net/
[2]http://www.music-ir.org/mirex/wiki

legato, articulations create 'muddied' results, especially with several notes in succession [3]–[5]. Pitched, non-percussive notes, like those from a flute signal, can be represented as a physical onset (when the amplitude peaks from zero), a perceptual onset (when the sound becomes audible), and/or a perceptual attack time (when the rhythmic emphasis is perceived) [6], [7], as seen in Figure 2, which shows a note of mid-range and moderate loudness level.

A. Soft Onsets and Vibrato

The long note onset of the flute (anywhere from 40-120 milliseconds, as seen in Figure 3) and the tendency towards natural manifestation of vibrato exposes challenges in properly detecting note onsets. Figures 3 and 4 exemplify two features that could cause indeterminacy in detecting note onset: the fundamental (F_0) and harmonic (F_N) frequencies, as well as long and/or soft note onset. The top portion of each figure is the waveform representation. The bottom portion is the spectrogram representation, where the color of the frequency dictates the energy (yellow is high energy at a particular frequency bin, and blue is low energy).

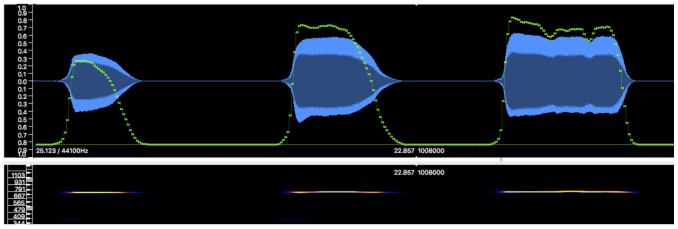

Fig. 3: Onset Profiles: Soft, Medium, Loud. Audio waveform (in blue) and RMS amplitude profile (in green).

"Vibrato associated frequency and amplitude modulation provides problems to traditional energy based onset detectors, which tend to record many false positives as they follow the typically 4-7 Hz oscillation" [4]. This range translates respectively to 250-147 milliseconds. Figure 4 shows various vibrato profiles in waveform representation (in blue) with an overlay (in green) of the root-mean-square (RMS) amplitude profile. Note the oscillatory nature of the RMS function in the rightmost waveform in Figure 4. This shows strong vibrato, whereas the middle waveform depicts a medium strength vibrato. The leftmost waveform has few oscillations and is almost steady state.

III. Using and Adapting Existing Algorithms

"Time-domain methods for producing onset detection functions are possible, but [the] most current techniques convert the signal to the frequency or complex domain" [8]. Iterating on and extending the work by Bello [6], [9] and Dixon [10], and constraining requirements to flute tudes, this section details the use and adaptation of several automatic audio onset detection algorithms. The following sections each provide a high-level overview of a given algorithm, its tuned variable parameter settings based on domain knowledge of flute signals, applied theory, and empirical testing, and its performance outcome.

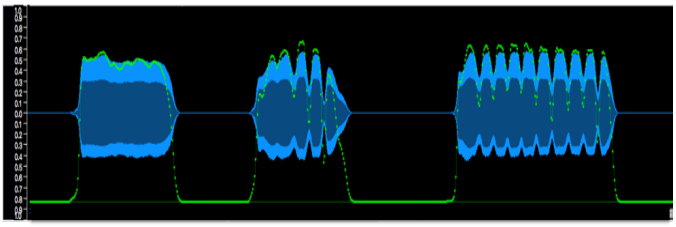

Fig. 4: Weak, medium, and strong vibrato profiles (left to right). Audio waveforms (in blue) and RMS amplitude profile (in green) depict the oscillatory nature of the signal.

Further information about each algorithm can be found in its associated reference.

A. Statistical Analysis

The tolerated *baseline assessment*, or ground truth, for each of the five experienced flutist's recordings are generated. The ground truth, which is subject to human perception, was compared to an algorithm's outcome. Statistical analysis between the established ground truth and annotations from the experienced flutist was run to determine an algorithm's performance. An algorithm's ability to discern whether or not an observed musical event is a proper note onset was under review. The goal is to achieve the highest number of true positives and the least amount of false positives and false negatives. These algorithms were analyzed based on a ratio of true positives to false positives and false negatives using the statistical operations: Precision, Recall, and F-measure, described below.[3] These evaluation metrics are commonly used in MIREX to assess an algorithm's viability.[4]

- Precision (P), or positive predictive value, is the number of correctly identified onsets (true positives) divided by all of the identified instances (true positives and false positives).
- Recall (R), or sensitivity, is the number of correctly identified onsets (true positives) divided by the properly identified instances (true positives and false negatives).
- F-measure (F) is the harmonic mean of precision and recall, as in Equation 1.

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{1}$$

If an algorithm's peak picking threshold was set too high or too low, then the computer was more likely to indicate a note onset where there is not meant to be one (false positive), or not indicate a note onset where there is meant to be one (false negative). The preferred outcome from an automatic algorithm would show a high number of correctly identified note onsets (true positives) and yield an F-measure closest to 1.0 of ideal.

[3]Much of this methodology is adopted from Dixon's research concerning onset detection [10].

[4]http://www.music-ir.org/mirex/wiki/2016:Audio_Onset_Detection

B. On Evaluating Offline Algorithms

"State-of-the-art onset detection algorithms are still far from retrieving perfect results, thus requiring human corrections in situations where accuracy is a must" [11]. The total number of correctly identified note onsets for this excerpt is 44. The manual markings provided by music researchers are similar, where 96% of the outcomes are within an acceptable tolerance of 40 milliseconds.[5] If an automatic algorithm is within this tolerance, then that algorithm can be considered a success. However, an automatic algorithm cannot be expected to perform better than a human would for detecting perceptual onsets, as the ground truth is currently our best representation, and therefore, any deviation from ground truth will not be reinforced in an automatic algorithm. It is unacceptable for a real-time analysis algorithm meant to convey performance characteristics to incorrectly identify note onsets. As such, maximizing the percentage of ideal will be the focus of this research when assessing approaches towards achieving stable onset detection.

C. Existing Onset Algorithms

The mechanisms by which these algorithms calculate note onset include: a variant of Fast Fourier Transform (to determine pitch by way of fundamental frequency - F_0), or an analysis in the temporal or spectral domain using low-level features (such as complex domain or broadband spectral energy). Each algorithm has been peer-reviewed at a renowned international conference, such as ISMIR. An F-measure of around 0.7 is considered successful in MIREX[6]. However, acquiring a rating closer to 0.90 would be more beneficial for real-time beat detection.

In order to maximize frequency resolution of the FFT, the relationship between the sampling rate of 44.1 kHz and an FFT bin size of 16384 samples and FFT hop size[7] of 8192 samples are the preferred windowing settings. The 32770/16384 window and hop size provide a fraction of the necessary proper note onsets (as the window is too large and misses infrequent events). Smaller window and hop size of, for example, 8192/4096 conversely provide too many note onsets. However, for musical features that change rapidly (such as those discussed below in the aubio and pyin algorithms), a window size of 1024 samples with hop size of 512 is preferred.

1) MIRToolbox: The offline MIRToolbox framework includes an extensive integrated collection of operators and feature extractors specific to music analysis [1]. MIRToolbox has a wide range of feature extractors available. It is an ongoing research project designed by Olivier Lartillot to determine "which musical parameters can be related to the induction of particular emotion when playing or listening to music" [12]. The *mironsets* function, a signal-based method, shows "a temporal curve where peaks relate to the position of note onset times, and estimates those note onset positions" [12], as such "a local maximum will be considered as a peak if its distance with the previous and successive local minima (if any) is higher than the contrast parameter" [12]. There are three optional arguments included with this function, including 'Envelope' (which computes an amplitude envelope), 'Complex' within 'Spectral Flux' settings (which computes the spectral flux in the complex domain), and 'Pitch' (which computes frame-decomposed autocorrelation, as well as the novelty curve of the resulting similarity matrix). The 'Spectral Flux' option computes the temporal derivative of the spectrum [12]. The 'Complex' option, adopted from [9], combines information from both the energy and phase of the signal. This means calculations are occurring in the temporal domain, rather than the frequency domain. Peaks in the onset detection curve profiles of the audio signal (signifying bursts of energy) correspond to note onset.

Table I displays the results achieved by analyzing the five flutists and the representation of the computer-translated musical score (CTMS) generated by MIDI using MIRToolbox's algorithm. The F-measure ranges from 0.33-0.48, with an average of 0.399 of ideal. The Precision values are closer to zero because there was a high quantity of false positives, which decreases the ratio of true positives to true positives plus false positives. This was possibly due to a sensitive peak picking threshold incorrectly marking vibrato oscillations as note onsets. The Recall values are closer to 1.0 because few false negatives are detected, resulting in a ratio of true positives to true positives plus false negatives approaching the ideal case of 1.0. This is means that soft note onsets were properly detected among all recordings. However, two of the drawbacks of this implementation is that it is only available offline, which does not suit the purpose of this research: real-time feedback, and, while it does correctly mark true positives, there are too many false positives.

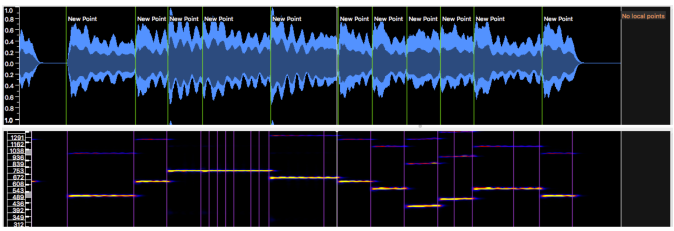

Fig. 5: This example shows how true note onsets (in green) relate to perceived note onsets (in purple) of MIRToolbox.

Figure 5 depicts true note onsets and perceived note onsets. The waveform and spectrogram representation are each overlaid with note onsets; the bright green annotations mark true note onsets from the baseline assessment and the bright purple notations mark note onsets observed by the algorithm. There are 11 true note onsets in this particular selection of music, however the algorithm observes 21. Of those 21, 11 are within the tolerated threshold of 40 ms. This shows that all the proper true note onsets are detected by MIRToolbox, however extra

[5]This tolerance is intended to improve on Dixon's work [10], where the accepted tolerance is 50 ms.

[6]http://nema.lis.illinois.edu/nema_out/mirex2016/results/aod/

[7]A 2:1 ratio of bin size to hop size is a commonly practiced standard https://ccrma.stanford.edu/ jos/parshl/Choice_Hop_Size.html

notes are also observed (most likely due to the sensitivity of the peak picking algorithm).

2) University of Alicante: Researchers at University of Alicante developed a signal-based interactive onset detection algorithm[8]. They approach "onset detection as a classification problem" [13], by using machine learning techniques to extract note onsets. After extracting audio features (such as energy, pitch, phase, or a combination of these) every few milliseconds, they implement a k-Nearest Neighbors classifier[9] to determine whether an event is an *onset* or a *non-onset*. This implementation is different from the algorithm within MIRToolbox in that it uses a machine learning technique to classify musical events.

The variable parameters were tuned to the system with a peak picking sensitivity of 20%, an FFT bin size of 16384, and an increment size of 8192 [11], [13]. Table I displays the results from University of Alicante's algorithm achieved. The average F-measure is relatively consistent among all five performances, at an average 0.75 of ideal. There is a similar quantity of false positives as there are false negatives, so the ratios of detected onsets to proper onsets (Precision and Recall) are within a standard deviation of 0.1. This means that the extracted audio features (currently unknown to the user) from each of the recordings used in the kNN exhibit similar tendencies and are consistently observed by this algorithm.

One example of how the algorithm's output compares to true note onsets is pictured in Figure 6. There are 11 true note onsets in this particular selection of music, however the algorithm observes 13. Of those 13, 10 are within the tolerated threshold of 40 ms. This means that, while some of the proper true note onsets are detected, there are two false positives present, lowering the percentage of ideal.

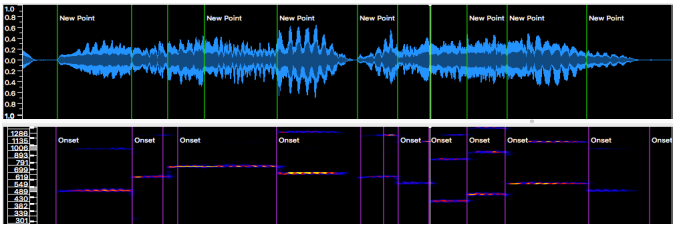

Fig. 6: This example shows how true note onsets (in green) relate to perceived note onsets (in purple) of Alicante.

3) QMUL: This onset detection algorithm plug-in was developed by Chris Duxbury, Juan Pablo Bello, Mike Davies, and Mark Sandler at the Queen Mary University of London. This signal-based method combines energy-based approaches (observing a signal's energy) and phase-based approaches (observing the deviations of the FFT state), which together form the complex domain. It includes an adaptive whitening component[10] that smooths the temporal and frequency variation in the signal so that large peaks in amplitude are more apparent by "bringing the magnitude of each frequency band into a similar dynamic range" [8]. By examining the spread of the attack transient distribution, as well as energy-based methods, the QMUL algorithm can "increase the effectiveness for less salient onsets" [2], such as long or soft flute note onsets. This algorithm calculates the likelihood an onset will occur within each frequency bin based on peaks in the complex domain, and uses a peak picking algorithm to mark an onset.

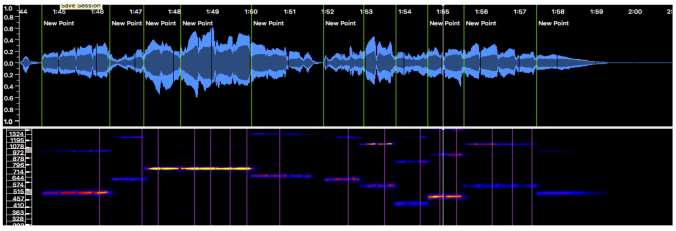

Fig. 7: This example shows how true note onsets (in green) relate to perceived note onsets (in purple) of QMUL.

Table I displays the results achieved using the Complex Domain algorithm. The variable parameters were tuned to the system with a sensitivity of 50%, an FFT bin size of 16384, an increment size of 8192, and a Blackman-Harris window shape [2], [8], [14]. The F-measure ranges from 0.34-0.62, with an average of 0.46 of ideal. The ratio of true positives to false positives and negatives is similar to that of the MIRToolbox algorithm. For Player 3, Precision is closer to zero than for any other player, which means it was difficult for the algorithm to correctly identify onsets. This could be due to the fact that Player 3's recording is somewhat softer in amplitude than the other recordings because the articulation used by the musician is legato and the musician exhibits deep vibrato, which is erroneously detected by the algorithm as an onset.

Comparing the results from QMUL to true note onsets is shown in Figure 7. There are 11 true note onsets in this particular selection of music, however the algorithm observes 16. Of those 16, only 3 are within the tolerated threshold of 40 ms. This is an example of how poorly this algorithm performs.

4) aubio: The aubio library was developed by Paul M. Brossier at the Centre for Digital Music at Queen Mary University of London. This real-time, signal-based onset detection algorithm functions similarly to the QMUL algorithm presented above. "A modified autocorrelation of the onset detection function is ... computed to determine the beat period and phase alignment [of the music]. Based on the detected period and phase, beats are predicted" [15]. This offline algorithm has two primary variable parameters: a threshold value from 0.01 to 0.99 (for peak picking) and an onset mode (for detection functions, including high frequency content, complex domain, energy, and spectral difference). As described in the

[8]http://grfia.dlsi.ua.es/repositori/grfia/otros/interactive-onset-detection.pdf

[9]The k-Nearest Neighbors algorithm is a method popularly used for classification, clustering, and regression analysis of points in closest proximity to one another.

[10]"A new method for preprocessing short-term Fourier transform phase-vocoder frames for improved performance on real-time onset detection" for slow onsets, like flute signals [8].

research by Dixon [6], using the complex domain is a preferred method for analysis, given the nature of the flute signal. The variable parameters were tuned to the system with an FFT bin size of 1024, an increment size of 512, a peak picking threshold of 0.5, a -50 dB silence threshold (reducing this allows low energy onsets to be observed), and a minimum inter-onset interval of 40 ms (two consecutive onsets will not be detected within this interval). A window size of 1024 and hop size of 512 was sufficient, due to the phase vocoder (which is used to obtain a time-frequency representation of the audio signal) [16]. The algorithm's superior ability to specifically observe long, slow note onsets [16] is a preferred feature of this algorithm. Changing the peak picking algorithm threshold lower or higher results in too many or too few onsets.

Table I displays the results achieved by analyzing the six recordings using aubio's algorithm. There are slightly more false results detected in aubio's algorithm compared to Alicante's, giving an average F-measure of 0.59 of ideal.

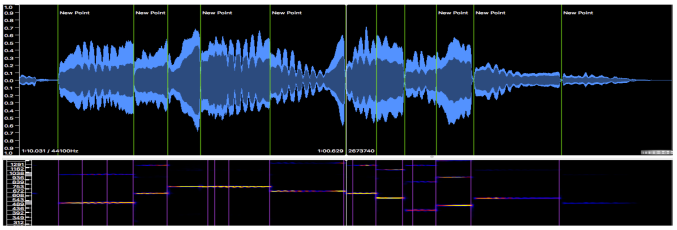

Fig. 8: This example shows how true note onsets (in green) relate to perceived note onsets (in purple) of aubio.

One comparison between ground truth and aubio's output is pictured in Figure 8. There are 11 true note onsets (marked in green) in this music selection, however the algorithm (marked in purple) observes 17. Of those 17, 11 are within the tolerated threshold of 40 ms. Similar to the MIRToolbox algorithm, all 11 true note onsets in this section of music are properly discovered.

5) pyin: The pyin algorithm is a real-time modification of the well-known, frame-wise *YIN* algorithm for fundamental frequency (F_0) estimation in monophonic audio signals [17] that produces pitch candidate probabilities as observations in a Viterbi-decoded Hidden Markov Model [18]. *YIN* is a term that "alludes to the interplay between autocorrelation and cancellation that it involves" [17] when estimating F_0. Pyin was developed by researchers Matthias Mauch and Simon Dixon at QMUL and extends the work by deCheveigne *et al.* [17]. It is different than the aforementioned algorithms in that it is designed to detect pitch, rather than explicit note onset, and is a probability-based method. It extracts multiple pitch candidates for given frequency ranges [19]. However, because the design of the algorithm annotates a timestamp along with the fundamental frequency estimation, it proves a valid contender for onset detection. This information is used to extrapolate note onset. The variable parameters were tuned to the system with an FFT bin size of 1024 samples, an increment size of 512, a *YIN* threshold distribution (which is a set of pitch candidates with associated probabilities) set to *uniform*[11], a suppression of low amplitude pitch estimates at 0.1 (which suppresses amplitudes below a certain value), and an onset sensitivity (equivalent to peak picking) of 0.7. The onset sensitivity changes how many onsets are detected.

Table I displays the results achieved by analyzing the six recordings using pyin's algorithm [18], [19]. The results from pyin are reminiscent of those from Alicante's algorithms, where the quantity of false positives is similar to the quantity of false negatives. The results make up approximately half of the properly detected onsets across all of the recordings. This algorithm has a better average F-measure (0.71 of ideal), similar to Alicante's results.

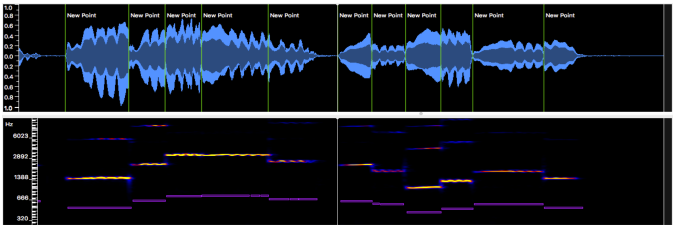

Fig. 9: This example shows how true note onsets (in green) relate to perceived note onsets (in purple) of pyin

One example of how the results from the pyin calculation compares to true note onsets is pictured in Figure 9. There are 11 true note onsets in this particular selection of music, however the algorithm perceives 16. Of those 16, all 11 true note onsets in this section of music are properly discovered.

D. Reflections on Onset Detection

The intention of this study is to examine the outcomes of previously tested note onset detection algorithms in order to observe which approach would best perform for solo flute signals. Several factors, based on the nature of the flute signal, impact the outcomes of the aforementioned algorithms. These factors include the long slope onset of the flute, legato (soft or gentle) articulation, and deep vibrato warble. During a legato articulation, the beginning of a note could be missed. Several of the automatic algorithms would incorrectly mark a warble in vibrato as a note onset if, for example, the parameters (such a peak picking threshold) were set too low - such as 0.25 instead of 0.5 or 0.75 - (as shown in Figure 10), or if the window size was too short. A false positive could be triggered by a strong vibrato.

For example, an algorithm's variable parameters are tuned such that it properly detects 75% of the *actual note onsets*. If the parameters are modified such that 100% of the *actual note onsets* are recognized, this results in more false positives and false negatives, as other features (such as vibrato) trigger incorrect onsets. This is unsuitable for real-time analysis of beat detection.

[11]Several F_0 candidates are obtained for each frame based on parameter distribution [18], [19]

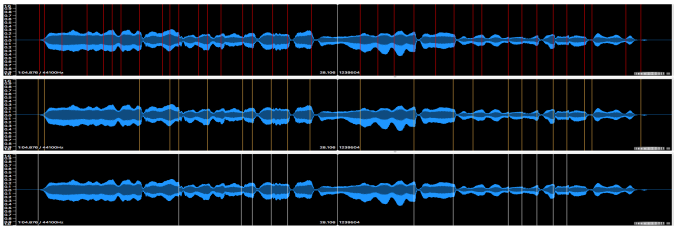

Fig. 10: This example shows low peak picking threshold (0.25) in red, medium peak picking threshold (0.5) in orange, high peak picking threshold (0.75) in white

TABLE I: Precision (P), Recall (R), and F-measure (F) for Various Automatic Algorithms. CTMS represents the computer-translated musical score.

	Player 1	Player 2	Player 3	Player 4	Player 5	CTMS
MIRToolbox	P:0.3071 R:0.8864 F:0.4562	P:0.2143 R:0.7501 F:0.3334	P:0.2123 R:0.8637 F:0.3408	P:0.2516 R:0.8864 F:0.3920	P:0.2616 R:0.7728 F:0.3909	P:0.3462 R:0.8183 F:0.4866
Alicante	P:0.8422 R:0.7274 F:0.7806	P:0.7047 R:0.7047 F:0.7047	P:0.7348 R:0.8183 F:0.7743	P:0.8050 R:0.7501 F:0.7766	P:0.6978 R:0.6820 F:0.6898	P:0.8650 R:0.7274 F:0.7902
QMUL	P:0.8828 R:0.3412 F:0.4922	P:0.8132 R:0.2958 F:0.4338	P:0.8132 R:0.2958 F:0.4338	P:0.8232 R:0.3185 F:0.4595	P:0.7157 R:0.2277 F:0.3455	P:0.8752 R:0.4775 F:0.6179
aubio	P:0.6401 R:0.7274 F:0.6810	P:0.3685 R:0.6366 F:0.4668	P:0.6831 R:0.6366 F:0.6590	P:0.4376 R:0.7956 F:0.5646	P:0.4784 R:0.7501 F:0.5842	P:0.5791 R:0.7501 F:0.5842
pyin	P:0.6605 R:0.7956 F:0.7218	P:0.5274 R:0.6593 F:0.5860	P:0.7858 R:0.7501 F:0.7676	P:0.7335 R:0.7501 F:0.7417	P:0.6925 R:0.6138 F:0.6508	P:0.8206 R:0.7274 F:0.7712

Table I collates the results gathered from the algorithms. These results correlate to the results from MIREX[12], corroborating the performances of the algorithms. The algorithms' performances are impacted by the difficulty in mathematically analyzing complex musical signals. Musical features such as a legato articulation and vibrato might be mathematically similar to a note onset, which is why some of the algorithms incorrectly identified note onsets. The lowest F-measures come from the MIRToolbox (Player 2 and 3) and QMUL (Player 3) algorithms, which means the audio features were not prominent enough to detect proper note onset. The highest F-measures come from the Alicante (Player 3 and 4) and pyin (Player 3 and CTMS) algorithms. It is interesting to note that Player 3 gives both the highest and the lowest F-measures of a given algorithm. This shows how the algorithmic approaches yield different outcomes for the same recording.

IV. Discussion on Onset Detection Algorithms

Despite best efforts to use automatic onset detection algorithms tailored specifically for flute audio signals, there still exists an unacceptable number of false positives and negatives (as represented in Table I). This is accentuated when attempting to perform analyses in real-time, as there is a trade-off between high performance and latency. If successive notes are repeated with a legato articulation, even an aural evaluation shows the events are difficult to distinguish. The higher performance algorithms (such as pyin) use frequency detection, however a delay exists when calculating the fundamental frequency in real-time (as seen in Figure 9). Additionally, vibrato could be incorrectly perceived as note onsets, and soft articulations could be missed if the peak picking algorithms are tuned such that all true note onsets are properly detected. If the peak picking threshold is tuned too low, there will too many note onsets. If the peak picking threshold is tuned too high, there will be missed note onsets. These algorithms have difficulty distinguishing *actual note onsets*, therefore, another approach, such as adding gesture signals, would be required for real-time observation of flute note onset.

[12]http://nema.lis.illinois.edu/nema_out/mirex2016/results/aod/summary.html

References

[1] O. Lartillot, P. Toiviainen, and T. Eerola, *Data Analysis, Machine Learning and Applications: Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[2] C. Duxbury *et al.*, "Complex domain onset detection for musical signals," *International Conference on Digital Audio Effects*, no. 1, pp. 6–9, 2003.

[3] S. Dixon, "On the analysis of musical expression in audio signals," *Storage and Retrieval for Media Databases*, 2003.

[4] N. Collins, "A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions," *Audio Engineering Society*, vol. 1, pp. 34–45, 2005.

[5] N. Collins, "Using a Pitch Detector for Onset Detection.," in *International Society for Music Information Retrieval Conference*, 2005.

[6] J. Bello *et al.*, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[7] N. Collins, "Investigating computational models of perceptual attack time," *International Conference on Music Perception & Cognition*, pp. 923–929, 2006.

[8] D. Stowell and M. Plumbley, "Adaptive whitening for improved real-time audio onset detection," *International Computer Music Conference*, 2007.

[9] J. P. Bello *et al.*, "On the use of phase and energy for musical onset detection in the complex domain," *IEEE Transactions on Signal Processing Letters*, vol. 11, no. 6, pp. 553–556, 2004.

[10] S. Dixon, "Onset Detection Revisited," *International Conference on Digital Audio Effects*, pp. 133–137, 2006.

[11] J. J. Valero-Mas, J. M. Inesta, and C. Pérez-Sancho, "Onset detection with the user in the learning loop," in *International Workshop on Music and Machine Learning*, 2014.

[12] O. Lartillot, *MIRToolbox 1.6.1 User Manual*. Department of Architecture, Design and Media Technology, Aalborg University, Denmark, 2014.

[13] J. J. Valero-Mas and J. M. Inesta, "Interactive onset detection in audio recordings," 2015.

[14] C. Duxbury, M. Sandler, and M. Davies, "A hybrid approach to musical note onset detection," *Computer Music Journal*, pp. 33–38, 2002.

[15] P. M. Brossier, "The aubio library at mirex 2006," *MIREX 2006*, p. 1, 2006.

[16] P. Brossier, *Automatic Annotation of Musical Audio for Interactive Applications*. Doctoral thesis, Queen Mary University of London, 2006.

[17] A. De Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[18] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[19] M. Mauch *et al.*, "Computer-aided Melody Note Transcription Using the Tony Software : Accuracy and Efficiency," *First International Conference on Technologies for Music Notation and Representation*, p. 8, 2015.