ON ONSETS ON-THE-FLY: REAL-TIME EVENT SEGMENTATION AND CATEGORISATION AS A COMPOSITIONAL EFFECT

Nick Collins
Centre for Music and Science
Faculty of Music, University of Cambridge
11 West Road, Cambridge, CB3 9DP, UK
http://www.cus.cam.ac.uk/~nc272/
nc272@cam.ac.uk

ABSTRACT

Compositional applications for real-time event segmentation are discussed. A causal real-time onset detector which makes onset data available as fast as possible is introduced, based on work by Klapuri, Hainsworth and Jensen and Andersen. This analysis frontend informs algorithmic cutting procedures which respect the events of the incoming audio stream. A further refinement stores events to particular buffers based on a coarse categorisation between snare, kick or hihat classes. Algorithmic composers running playback of these buffers thereby respond to changing timbral events of a live feed from an instrumentalist or other audio source. The use of an onset detection threshold to create abstracted rhythms based on some existing source is further examined.

Keywords: Onset Detection, Audio Capture, Real-time Segmentation, Categorisation

1. INTRODUCTION

The automatic segmentation and labeling of audio events has many applications from content retrieval to source sensitive sound processing. Practical attempts often employ signal features including the spectral centroid, zero crossing rate or MFCC coefficients and statistics tracking these values over the time course of the sound. Rossignol et al. [16] give a system that characterises signals on three scales, *source* differentiating speech and music, *feature* measuring such descriptors as harmonicity or presence of vibrato and *note/phone* segmenting signals into short subsecond events based on nine features. Current investigations into the classification of sounds can involve very large sets of features such that exhaustive search for the optimal subgroup is computationally intractable [7]. Spotting events requires reaction to to the perceptually critical segment boundary of the onset [20]. Many onset detection schemes have now been proposed [1, 3, 5, 9, 19]. The offset at the end of an acoustic event is usually less reliably marked [19], though the character of the onset can also vary from wideband transient to a smooth envelope, depending on the source. The effectiveness of event detection algorithms thus vary with the signal to be tracked and the assumptions taken about the source. It is quite possible in human hearing that selection is taken from a number of detection models based on peripheral evidence of the spectral content and resolved immediate stylistic characteristics, in accord with maximising information. It is assumed herein that such higher level management of low level algorithms is sidestepped.

Onset detection work is most effective against monophonic sources, the case of polyphonic audio providing a well known problem in auditory scene analysis. Whilst some progress may be possible through independent component analysis and the like, Scherier [18] criticises the dream of perfect segregation of streams in polyphony, rather seeking a more human goal of '*understanding without separation*'. When multiple instruments are involved, overlap of parts makes the act of segmentation without component extraction guaranteed to cut a mixture of transients and steady states. There is a compositional awareness of this, common for example in the fugal forms of Bach, where the entry of parts is staggered, and the counter melodies constructed to to aid differentiation of onsets by avoiding unison events. Onsets are thus accessible, and a rhythmic aggregate may be the cognitive resultant, but the overlap of sounding objects forbids perfect separation. Pragmatically, audio can be segmented and extracted as best possible, even if some blends of decay and new onsets result. It will be argued that transient information (the likely basis for the cuts) will take precedence in perception, and interesting compositional effects are obtained.

The focus of this paper is in live performance, where a (typically monophonic) acoustic instrument or a (polyphonic) live band or ensemble is being captured and manipulated, and segmentations are performed on-the-fly. The ideal is to react as quickly as possible, though some delay for analysis of the captured events my be necessary. Conversely, no faster than real-time look ahead is possible, as in streaming an existing soundfile from disk. The time duration of the extracted events is usually under one second, fitting into the *note/phone* scheme of [16], but large enough to allow perceptual integration (over 100mS). This level is suitable for rhythmic rate manipulation in processing, and is the *sound object* time scale of Roads' taxonomy [15]. Segmenting an incoming audio stream allows

the extracted sound events to be individually processed, and their reuse outside the original context.

The implementations described in this paper were undertaken by writing native code and C++ plugins for the SuperCollider3 (SC3) audio programming language [12] (http://sourceforge.net/projects/supercollider). The orientation of this language towards realtime performance combining signal processing and algorithmic composition makes it an attractive platform for the research; it is also open source and cross platform (Linux, Mac OS X).

Section 2 provides a mathematical description of the onset detection schemes leveraged. Section 3 describes a technology for live algorithmic audio cutting, integrating onset detection. This is the basis for an on-the-fly segmenter with a ready made selection of compositional algorithms. An experimental program for realtime categorisation is introduced in section 4 and further compositional extensions explored in section 5.

2. ONSET DETECTION METHODS

A number of onset detection functions are being investigated to support segmentation effects. Anssi Klapuri [9] pioneered the use of the relative difference for peak detection given an amplitude envelope signal $A(t)$. His psychoacoustic motivation was the relation to the Weber fraction $\Delta I/I$ for discrimination of intensity changes, for his function $W(t)$ takes the form:

$$W(t) = \frac{\frac{d}{dt}(A(t))}{A(t)} = \frac{d(log(A(t)))}{dt} \quad (1)$$

.

If $A(t)$ is below a certain amplitude, $W(t)$ is taken as zero.

Whilst Klapuri first applied this on the outputs of a 21 band envelope extractor and combined results across bands with a loudness model[1], Hainsworth [5] [6, page 128] introduces an equivalent construction[2] in the context of distance measures between FFT frames for chord change detection:

$$d(k) = log_2(\frac{|X_n(k)|}{|X_{n-1}(k)|}) \quad (2)$$

$$\sum_{k=\alpha}^{\beta} max(d(k), 0) \quad (3)$$

where α and β define lower and upper limits for a particular subset of bands and $|X_n(k)|$ is the magnitude of the k^{th} bin for the n^{th} frame of spectral data. Hainsworth selects 30Hz to 5KHz as his range on the basis of quality of harmonic information for his applications, though this is also of course the area of greatest sensitivity of the ear. A generalisation of this would weight the different bands, perhaps using Fletcher Munson contours based on the intensity of the input for a psychoacoustically relevant model, and perhaps in a way learnt for specific tracking tasks. The weighting may be selected so as to focus attention on particularly salient bands for an identified timbral profile. Further, such a weighting might bias detections to a desired frequency range, as in tracking only low frequency energy impulses in a bass drum or bass guitar rhythm, though this may also be achieved by pre filtering of the input.

$$d(k) = w(k)(log_2(\frac{|X_n(k)|}{|X_{n-1}(k)|})) \quad (4)$$

In comparison to the multiplicative difference, the additive difference is also an option:

$$d(k) = w(k)(|X_n(k)| - |X_{n-1}(k)|) \quad (5)$$

Where $w(k)$ is usually a constant 1. The onset detection feature of High Frequency Content (HFC) can be expressed where $w(k) = k^2$ as defined by Jensen and Andersen [8], $w(k) = k$ as per Masri and Bateman [11] or a generalised $w(k) = k^\gamma$ where γ is to be optimised.

Figure 1 gives a comparison of different onset detection methods on rhythmic polyphonic audio (some 'intelligent dance music', compressed and with many hard transients). 1024 point FFTs were taken with an overlap of two, on a spectral range of 30-20000Hz. The first row gives the source, the second the detection function for the relative difference measure, the fourth for the Jensen and Andersen version of HFC. Rows three and five show a squared difference function of the respective detection functions which highlights their peaks. Note that the noise floor is a little higher for the relative difference function, and that there are differences of opinion of location and strength of spectral changes between the two functions. As a contrast, Figure 2 shows the same detection functions acting on a solo violin piece with many dynamic contrasts. The Hainsworth/Klapuri spectral amplitude ratio method shows some superior detections here, particularly for the slower and softer attacks. This is in no way a proof of the general properties of these detection schemes, which vary in their effectiveness based on the subject audio to analyse. It is envisaged that different onset detection algorithms may be required for different circumstances of employment, optimising for a particular performance environment and musical collaborator.

The Hainsworth FFT version of Klapuri's idea and the Jensen and Andersen HFC feature were implemented as phase vocoder UGens for SuperCollider 3. For efficiency, the SC3 implementation uses an overlap of 2 for a 1024 point FFT, the 86 or so frames a second providing a resolution of under 12 mS. Hainsworth runs his 2048 point FFT calculations at an overlap of 8, giving around 170 frames per second, whilst Klapuri downsamples his signals to about 200 samples per second. Since the detection functions will pick up on a spectral change early on in an event (near the physical onset rather than the perceptual), the onset can be stored quickly and the offset calculated

[1] Klapuri tracks loudness in each band according to a psychoacoustic model, using this to form an overall loudness measure of the event from which to assess a detection by thresholding.

[2] For a difference equation approximation to the derivative, $\frac{d(log(A(t)))}{dt} \approx log(A(t)) - log(A(t-1)) = log(\frac{A(t)}{A(t-1)})$

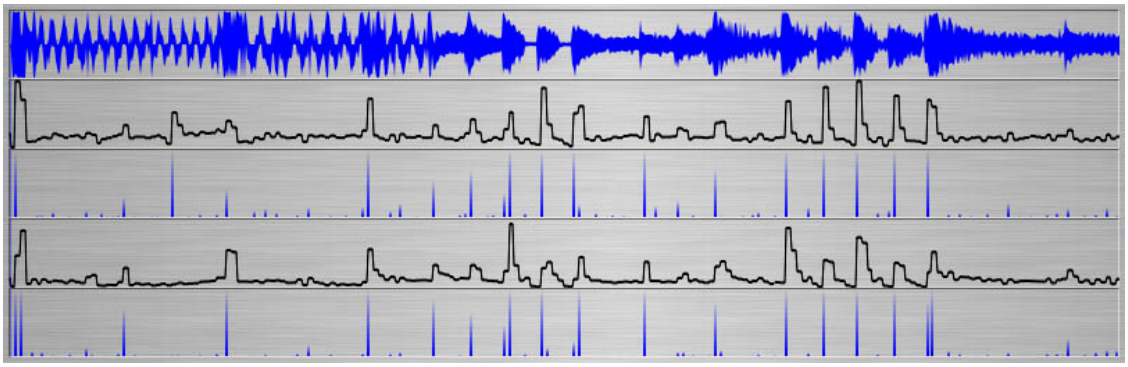

Figure 1. Comparison of relative difference (upper) and additive difference with k^2 weighting (lower) onset detection functions for a Squarepusher dance music track

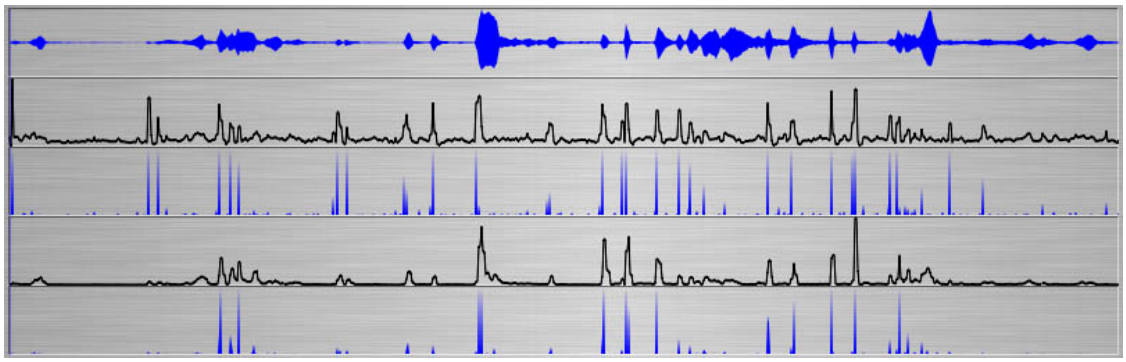

Figure 2. Comparison of relative difference (upper) and additive difference with k^2 weighting (lower) onset detection functions for a John Cage solo violin piece

as the stream continues to arrive. As long as the playback head is kept away from the record head, this allows immediate reuse or processing of events within an FFT frame of their detection. The last zero crossing or energy minima can be stored to keep track of a sample starting position for new events, though this is less effective for polyphonic or slurred monophonic audio.

3. BBCUT AND ONSETS

BBCut [2] is an extension library for SuperCollider specialising in realtime algorithmic audio cutting, available free under the GNU GPL from http://sicklincoln.org/code. Real-time computer generated splicing of audio material is provided in procedures inspired by a variety of styles, including dance music production drum programming and breakbeat cutting (drum and bass, intelligent dance music), thrash drumming, recursive cutting, change ringing permutation patterns and other compositional algorithms. BBCut provides a separation of the algorithmic composers that decide upon cuts from the rendering of the cuts themselves, such that the same composition code can be plugged into a cutter of incoming audio streams, or a cutter of fixed buffers; as software engineering, this encourages code reuse.

Whilst the naivest mode of cutting is to assume even slicing of the source material, there are options to adopt knowledge of the permissible onsets into a source. Onset positions are critical to audio cutting, where the paradigm is to jump around the source to make permutational use of the material rather than subsist on rigid linear playback. This generative reuse of the source material will work effectively if the source has been properly segmented into events such that there are no transient discrepancies acting as rhythmic confounds. This is conceptually the same as positioning a read pointer into a soundfile or buffered memory of a live stream, such that the start position at the beginning of a new cut lines up sensibly with events within that target[3]. Enveloping may be applied to avoid clicks and to smooth out transitions, but if natural cutting points can work without such ramping, so much the better.

How does BBCut use segmentation information for the source to be spliced, when asked to source audio data for an arbitrary length output slice? The simplest option is to restrict any choice of onset to the start positions of the segmented events of the source, and to only allow one event to be referenced at a time, such that playback is of that event alone, whatever the desired output length.

A more indirect option associates each known onset in the source to a quantised position. This location is determined with respect to an imposed rhythmic template of the algorithmic cutting[4]. A likely assumption would be

[3]There may of course be conflicts between the imposed rhythm of the composition and the recorded rhythm of the sample source.

[4]The degree this will respect or pervert the rhythmic frame of the

that the source is a certain length in beats and of fixed tempo and time signature. The order of the onset events is preserved in such a quantisation. The advantage is that extractions of material which are of medium length (say a few beats), and may take in more than one segmented event from the source, can guarantee playback that does not clash the rhythm of the source with the rhythm of the generative cutter. Playback of successive source events occurs at separations taken from the quantise positions, whilst the audio is still read from the original known onset positions. The rhythmic template, with whatever notion of groove or swing, can be adjusted, and the quantise position of the source events updated to reflect this. In computational terms, this requires a further data array for the quantised locations; search code finds the set of quantised event starts within this array falling within a cut of some duration from a given starting point. Knowing the indices within the quantise position array tells us immediately the indices for the events in the true onset position array for the source. These events can be rendered at the appropriate time for the quantise array so they fit in with the imposed template, reading the data from the source.

To assist in discovering onsets in offline preparation of source material, a GUI for onset detection (Figure 3) was introduced in BBCut1.3, with a number of options for the detection algorithm, all of which run in realtime. Adaptations of the Hainsworth and Jensen and Andersen methods described above are provided along with a simple RMS (root mean square) amplitude derivative detector. Since the two former FFT based routines have been built into the SuperCollider distribution independent of BBCut, they are also available as general purpose realtime detectors; the GUI is just a helpful frontend. Discovered segments can be played back individually in the GUI via keyboard shortcuts and misdetects deleted or moved.

In live performance, a buffer can be allocated to receive captured audio, and event detection fully automated. As the audio is recorded, the onset detection simultaneously runs, noting times of triggers of the detection function as event onsets, and disallowing multiple detects within 50mS of each other. The event offsets are taken as a maximal fixed duration from the onset or coinciding with a new onset. This is sufficient for a single fixed capture to make the new events available via the technology already described.

As a further refinement, a continuous stream of input audio can be continually analysed for events whilst recording to a limited memory circular buffer. If the onset data is held in a list and the capture continuous, since the write pointer position is known, the list can be continuously updated with onset information, and all read pointer access will stay up to date with the changing buffer contents. Where this is required, updating the quantise array as well will keep the cut renderer in step with the incoming stream.

source depends on further assumptions or knowledge on the captured audio, obtained possibly through beat/metre induction principles.

4. CATEGORISATION ON-THE-FLY

The best set of signal descriptors for classification of sound can depend on the categories of sound to be judged. For general sound classification, Peeters and Rodet [14] describe the CUIDADO system which is open-ended in features and user customisable in the type of sounds to classify, discriminating a relevant subset of features for a particular classification task. Categorisation of percussive sounds is tackled by Paulus and Klapuri [13] using a probabilistic model based on ten signal features, and Herrera et al. [7] exploring over two hundred.

Without tackling the best selection of features, an on-the-fly categoriser was built as an experiment in compositional application for event segmentation based on the onset detector already introduced, . The goal of this prototype is categorisation of incoming sound events as soon as possible, into one of three classes, notionally being kick, snare and hihat percussive sounds. The single feature initially used for classification in prototyping is the average spectral centroid bin:

$$\frac{\sum_{n=0}^{L}\sum_{k=0}^{N/2} k|X_n(k)|}{L} \tag{6}$$

Where there are L frames of an N point FFT in a given event.

Hiding certain technicalities based on blocksize calculation, pseudocode for an on-the-fly categorisation algorithm is presented in figure 4.

A SuperCollider UGen, CaptureCategorise, was written in C to implement this. The UGen has inputs for the threshold of detection, and to choose the boundaries for the feature determining classification. Defaults were average centroid bin below 90 for a kick, below 120 for a snare and a hihat above that. This was sufficient to demonstrate some live (vocal) beatbox control, with captured buffers being played back in a basic generative drum beat, and the appropriate contents continually overwritten when a new event was detected.

A more robust system would entail learning from a database of examples, even an online learning process during performance, to discover a relevant feature space for discrimination. Still, the exploration of further basic features provides some immediate compositional dividends.

5. FURTHER COMPOSITIONAL APPLICATIONS

In the course of an exploration of causal real-time onset detection functions, the author had recourse to listen back to sources, whilst simultaneously triggering beeps at the detections. It was noted that by changing the non-adaptive threshold of the detection function, a series of abstracted rhythms could be generated from the source. Shifting threshold gives a complexity parameter for the generated rhythms. The source may then be hidden, and the detections used to trigger arbitrary sound events. A form of onset detection cross synthesis can take place when one

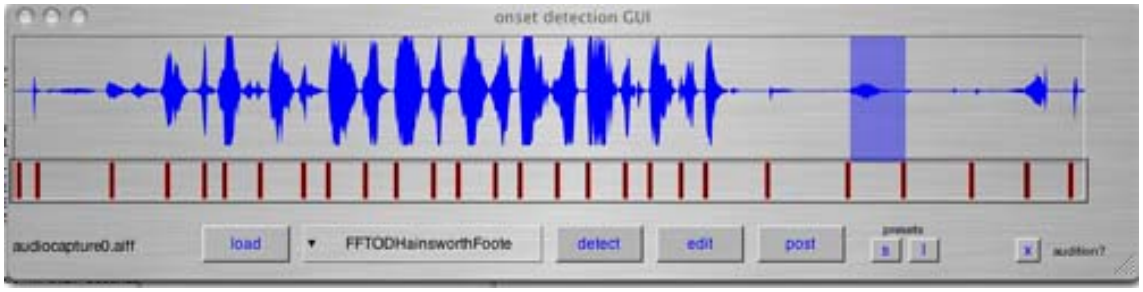

Figure 3. BBCut Onset Detector GUI

```
for each FFT frame {
        store last time domain zero crossing
        if(recording event) {
                calculate running spectral centroid, power of frame

                if(eventlength>MAXLENGTH or power<MINPOWER) {
                        finish recording to temporary buffer up to last zero crossing
                        copy event data to one of three buffers based on the
                        time averaged spectral centroid
                        }
                else store frame data to temporary buffer and increment eventlength
                }
        else if(onset detected) start recording event from last zero crossing, initialise eventlength
}
```

Figure 4. Pseudocode for an on-the-fly capture and categorise algorithm

source provides the trigger rhythm, and events extracted from a second are triggered.

As one example of this technique, the capture and categorise process ran on one input stream, classifiying input events to three buffers. These buffers were played back using rhythms generated by thresholded onset detection from three separate source loops.

In an aesthetic sense, misdetections increase the abstraction and the less accurate onset detection functions and less salient signal features may still be appropriated for compositional purposes. In the on-the-fly categorisor, miscategorisations can provide some stimulating results!

6. CONCLUSIONS AND FUTURE WORK

Whilst an imposed rhythmic framework has provided the skeleton for reuse of sound objects in this work, the incorporation of pulsation levels and other musical knowledge extracted from a cutting target should provide a more powerful system yet. Event detection is a first stage in beat induction, providing readily available inter onset data for histogramming or autocorrelation, though the leap to the symbolic stage is not a necessity, as Scheirer's model proves [17]. A model which provides a higher level rhythmic framework empowers some interesting processing options, as Gouyon et al. prove [4].

A more systematic testing of the onset detection algorithms is a necessary next stage, optimising some of the parameters indicated in section 2. It is anticipated that specific compositions, working with particular instrumentalists, may demand the use of a particular detection algorithm. The categorisation process can certainly be made more robust, and a key issue is the best personalisation to the sound world of a collaborating musician. Databasing of a timbral space during an interaction would be a worthy pursuit, which is well within reach if the composer specifies the feature space in advance. Captured events can be further analysed for the perceptual centre [10] so as to make sure that the physical start is appropriately scheduled. Another potential problem to be dealt with is the accidental capture of double strikes and other potential conflicting rhythmic confounds in single events which disturb the rhythmic flow of the output.

This paper is a stepping stone towards a sensitive system for audio capture and processing. A number of compositional applications of real-time onset detection have been highlighted.

7. ACKNOWLEDGEMENTS

This research is supported by AHRB grant 2003/104481. Many thanks to Juan Bello, Kristoffer Jensen and Steven Hainsworth.

8. REFERENCES

[1] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and S. B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 2004.

[2] Nick Collins. The BBCut Library. In *Proc. Int. Computer Music Conference*, pages 313–6, Goteborg, Sweden, 16-21 September 2002.

[3] Chris Duxbury, Juan P. Bello, Mike Davies, and Mark Sandler. Complex domain onset detection for musical signals. In *Proc. Digital Audio Effects Workshop (DAFx)*, 2003.

[4] Fabien Gouyon, Lars Fabig, and Jordi Bonada. Rhythmic expressiveness transformations of audio recordings: swing modifications. In *Proc. Digital Audio Effects Workshop (DAFx)*, 2003.

[5] Stephen Hainsworth and Malcolm Macleod. Onset detection in musical audio signals. In *Proc. Int. Computer Music Conference*, pages 163–6, 2003.

[6] Stephen W. Hainsworth. *Techniques for the Automated Analysis of Musical Audio*. PhD thesis, University of Cambridge, 2004.

[7] Perfecto Herrera, Amaury Dehamel, and Fabien Gouyon. Automatic labelling of unpitched percussion sounds. In *AES 114th Convention*, Amsterdam, March 2003.

[8] Kristoffer Jensen and Tue Haste Andersen. Real-time beat estimation using feature extraction. In *Proc. Computer Music Modeling and Retrieval Symposium, Lecture Notes in Computer Science*. Springer Verlag, 2003.

[9] Anssi Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, pages 3089–92, 1999.

[10] Stephen M. Marcus. Acoustic determinants of perceptual center (p-center) location. *Perception and Psychophysics*, 30(3):247–56, 1981.

[11] Paul Masri and Andrew Bateman. Improved modelling of attack transients in music analysis-resynthesis. In *Proc. Int. Computer Music Conference*, 1996.

[12] James McCartney. Rethinking the computer music language: SuperCollider. *Computer Music Journal*, 26(4), 2002.

[13] Jouni Paulus and Anssi Klapuri. Model-based event labelling in the transcription of percussive audio signals. In *Proc. Digital Audio Effects Workshop (DAFx)*, 2003.

[14] Geoffroy Peeters and Xavier Rodet. Automatically selecting signal descriptors for sound classification. In *Proc. Int. Computer Music Conference*, 2002.

[15] Curtis Roads. *Microsound*. MIT Press, Camb, MA, 2001.

[16] S. Rossignol, X. Rodet, J. Soumagne, J.-L. Collette, and P. Depalle. Automatic characterisation of musical signals: Feature extraction and temporal segmentation. *Journal of New Music Research*, 28(4):281–95, 1999.

[17] Eric D. Scheirer. Tempo and beat analysis of acoustic musical signals. *J. Acoust. Soc. Am.*, 103(1):588–601, January 1998.

[18] Eric D. Scheirer. Towards music understanding without separation: Segmenting music with correlogram comodulation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999.

[19] Leslie S. Smith. Sound segmentation using onsets and offsets. *Journal of New Music Research*, 23:11–23, 1994.

[20] Joos Vos and Rudolf Rasch. The perceptual onset of musical tones. *Perception and Psychophysics*, 29(4):323–35, 1981.