Aaron Rourk
Prof. Devaney
Computational Musicology
Fall 2018

Phrase Detection

Abstract
The goal of this project was to create a software device (using the Max visual programming environment) capable of detecting phrases in monophonic, melodic improvisations. The device, which utilizes a combination of automatic and manual control, was applied to audio files that fit the above description in addition to files that are better described by other categories (polyphonic music, speech, alternative audio sources, etc). While the intention was primarily to create a functioning phrase detector to use with monophonic, melodic improvisation, the resulting device is quite capable of producing interesting creative results on any kind of audio input.

Inspiration

This project was inspired by the work of projects like Robert M. Keller's "Impro-Visor", the *Experiments in Musical Intelligence* of David Cope and Belinda Thom's "BoB". These projects all involve the generation of unique musical phrases in a number of given styles. The device I created was intended to be a step, for me, toward a project of this magnitude. Instead of tackling the whole problem of musical phrase generation based on input, I narrowed my focus to the realm of phrase detection for live audio signals. While my initial focus was offset detection, I quickly developed an approach that utilized a filtering. The idea is that one could use the phrase information they receive from the device (which uses a mixture of automatic and manual control), to generate unique phrases based on audio input of their own by playing back the sliced phrases using MIDI pitch input.

Implementation

The following section will describe the controls of phrase-sampler.amxd by column, proceeding from left to right.

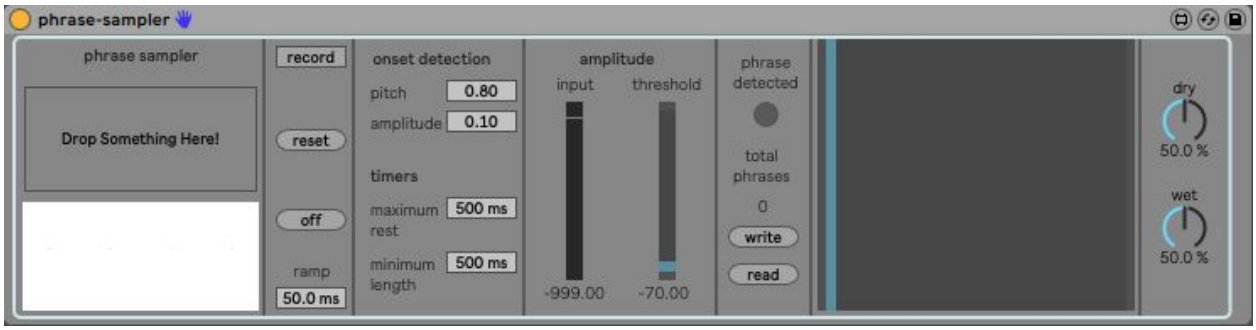

Figure 1
The phrase-sampler.amxd Max for Live device, as it appears in Ableton Live 10.

MIDI Input
Because this device is classified as an Max for Live Instrument (as opposed to an Audio or MIDI Effect), it is equipped with a MIDI input. Incoming MIDI pitches are used to trigger playback of specified phrases (0-127), additionally triggering random pan values for the current phrase playback voice as well cueing the voice to ramp up its volume and ramp its playback speed to the speed of the original recording.

<u>Column 1</u>
"Drop Something Here!"
Click and drag an audio file to be analyzed or played back in this space.
**The blank rectangle visualizes the audio file once it is dropped in.*

<u>Column 2</u>
"record"
Triggers playback of the dropped audio file and begins counting phrases, based on onset and phrase detection settings.

"reset"
Erases the memory.

"off"
Turns off all phrase playback voices.

"ramp"
Sets the time, in milliseconds, that it takes for 1) the pan to reach its new randomly chosen value, 2) for the phrase playback voices to reach a playback speed of 1 times the original (from 0, or stopped) and 3) for the playback voice audio to reach an RMS value of 100, or 1.

Column 3
“pitch”
Sets the amount, in MIDI notes, that the pitch may fluctuate before an onset is detected.

“amplitude”
Sets the maximum amount, in 0.-1. RMS values, that the amplitude can change before an onset is detected.

“maximum rest”
The maximum amount of time that may pass while the input amplitude is below the amplitude threshold before an onset is passed through to the “minimum phrase length” stage.

“minimum phrase length”
The minimum amount of time, in milliseconds, that must pass before a new onset is allowed to trigger the phrase counter, thereby storing a phrase number/point-in-recording pair in the memory.

Column 4
“input”
The current amplitude of the input audio (playing from the dropped audio file with “record” is selected) in decibels (dB).

“threshold”
The minimum amplitude, in decibels (dB), that the input audio must be in order for an onset to pass through to the “maximum rest” stage.

Column 5
“phrase detected”
The LED circle notifies the user of a phrase detection.
All onsets that are passed through the gates described in Column 3 and Column 4 trigger a two-step process: 1) a counter is advanced and 2)

This generates a list comprising of the phrase number (0…∞) followed by the point-in-recording (ms)

which can be written to disk using the “write” button, described below.

“total phrases”
The total number of phrases detected so far.

“write”
Writes the contents of the phrase memory (phrase number, point-in-recording) to the hard drive.

“read”
Reads a phrase memory file for use with a specific audio file, which can be dragged-and-dropped into the device for playback.

Column 6
**The drawbars in this column represent the pan for each of the 6 phrase playback voices.*

Column 7
“dry”
the amount of dry signal (the dropped audio recording, if “record” is den, as an RMS value) that is passed through to the output

“wet”
the amount of dry signal (from input MIDI pitches, as an RMS value) that is passed through to the output

instructions

initial phrase capture

1) load recording
2) press “record”
3) monitor levels (onset pitch, onset amplitude, maximum rest, minimum length, amplitude threshold) while watching “onset detected” LED,
4) press “write” to write phrase data to text file
5) play back tracked phrases with MIDI notes (0-127)

playing back pre-existing phrase captures

1) drag and drop desired recording
2) click “read” and select corresponding text file
3) play back tracked phrases with MIDI notes (0-127)

What is a phrase?
For the purposes of this task, a phrase was to be the musical equivalent of a sentence. Not a phoneme, not a syllable, not a word; a sentence, or at least a significant portion of thereof. For instance, if "I am going to walk my dog around the block" was considered the 'phrase' in question, any sensible variation thereof would be acceptable for the the device to record as a phrase ("I am going," "I am going to walk," "I am going to walk my dog").

Design
In order to obtain useful results, the user must monitor, in real time, several aspects of the device namely the properties called "maximum rest", "minimum phrase length" and "threshold", as described above. Because these controls are monitored manually by a human user, the results will be biased toward the users ear depending on what they think constitutes a phrase in the given context.

There are other factors that make phrase detection a tricky concept

1) The amplitude content that comprises a phrase changes drastically between genres/styles/people.
2) The intention of the performer is of huge importance
3) The way in which the listener constructs the narrative of all the sound they have heard so far in a given performance affects how they define a phrase.

<u>Advantages</u>

- able to process audio in real time
- can be used as a real-time, buffer-based audio effect

<u>Disadvantages</u>

- slow to process an entire file (has to be done in real-time)
- user has to manually monitor settings

<u>Future Additions</u>

- presets for different phrase types (long, short, dynamically varied, etc)

<u>Applications</u>

- as a way to generate new material from recorded sounds (as an instrument in the way that a turntable is an instrument)
- as a live audio effect (buffer-based delay, phrase delay)
- the real-time capabilities are what makes it an exciting creative tool

Data & Results
Since this was a creative project, some of comes in the form of audio created by re-playing the

Folders
"audio"
Contains the sample audio files used to create the tables, playback pieces and plots.

"phrase tables"
Contains text files containing the phrase number/point-in-recording (ms) pairs for each sample file

"phrase playback pieces"
Contains audio recordings of creative pieces made with the device, using the phrases deciphered from altofluteCm.wav, clarinetCm.wav, fretlessbassCm.wav and clownmusicbox.wav. The piece derived from the phrases of "clownmusicbox.wav" was created in real-time, so we can hear the device functioning as a delay. This is especially noticeable at the beginning.

Conclusions
As is noticeable when playing back the detected phrases (following the playback instructions above), the device does a fairly good job of detecting phrases appropriately when it receives manual assistance from the user, as described above.

The process on the whole would probably be more accurate to begin with annotated MIDI files (with phrase beginnings/ends, pitches accounted for), then examine the data and ascertain information like mean phrase length, % of phrases within several time ranges (0-100ms, 100-500ms, 500-1000ms, 1000-2000ms, etc), and additional categories. Then the program would use the culled information to detect phrases on its own. This data could then also be used to generate new, synthesized phrases in the style of the input MIDI file.

Alternatively, instead of using audio to find the phrases, one could make a table comprising of onset-detection points (ms), amplitude and use this information to cull phrase information.

However, considering all of these possible alternatives and next steps, the device is very capable of creating interesting musical - or at least *sonic* - results. I will certainly be using this as a part of my artistic process.