Trigger and Pitch Sequencer for Euclidean Rhythms

Christopher Latina
Georgia Institute of Technology
840 McMillan St.
Atlanta, Georgia 30332
chris.latina@gatech.edu

ABSTRACT

As a mode of interaction, this paper introduces a sequencer interface and visualizer for pitched euclidean rhythms. Inspired by Chicago footwork producers like DJ Rashad and Godfried Toussaint's 2005 paper, "The Euclidean Algorithm Generates Traditional Musical Rhythms", this program presents an interface for sequencing rhythms generated using the Bjorklund algorithm. Along with the timing algorithm, the interface includes a pitch sequencer and stereo-phase scope visualization. Intended for live performance environments, the euclidean sequencer enables users to improvise and create satisfying rhythms that explore the subtleties in discrete compositional approaches informed by visual and aural feedback.

Keywords

Euclidean Rhythms, MIDI Controllers, SuperCollider, Interactive, Visualization, Stereo-Phase Scope, Oscillographics, Bjorklund Algorithm, Footwork, Pointillism

1. INTRODUCTION

The Bjorklund algorithm is a technique to order timed pulses. A number of existing interfaces employ this algorithm for both hardware and software. On the hardware front, Mutable-Instrument's Grids and Yarns modules [2] and Rebel Technology's Stoicheia module [7] make use of euclidean sequencing for the eurorack format. With these modules, pattern sequences are controlled by a knob that updates instantaneously. This becomes problematic when performing with the tool because error-prone transitions can be clearly heard.

The concept behind the trigger sequencer is simple; allow the user to set a sequence using only two numbers, namely the sequence length and the fill amount, and then only trigger the pattern to update when desired while maintaing quantization. The newest iteration of the software also includes a pitch sequencer, tempo control, and stereo-phase scope visualization, which exploits the synthesis techniques behind each voice. The digital instrument can be used in either studio or live performance environments with an output that is reliable and captivating.

2. EUCLIDEAN RHYTHMS

As described in Euclid's *Elements*, the Euclidean algorithm finds the greatest common divisor between two numbers

[1 1 1 1 1 0 0 0]
[10] [10] [10] [1] [1]
[101] [101] [10]
[1 0 1 1 0 1 1 0]

Figure 1: Bjorklund(5,8) from Godfried Toussaint's paper, "The Euclidean Algorithm Generates Traditional Musical Rhythms"

through recursive subtraction. Bjorklund's algorithm follows the same structure but focuses on arranging quantized pulses in a manner that is as evenly distributed as possible [8]. Toussaint clearly demonstrates the connection between the Bjorklund algorithm and the Euclidean algorithm in his 2005 paper [8]. He describes how the Bjorklund algorithm as fundamentally Eudlidean and coins the generated patterns as Euclidean rhythms. Toussaint describes traditional rhythmic patterns in terms of euclidean fill and sequence lengths. In essence, the output of the algorithm is discrete and reproducible. This version uses SuperCollider's implementation of the Bjorklund algorithm by Fredrik Olofsson installed as a quark (or add-on). This implementation proved to be more stable and easily available than a homebrewed implementation. To install in SuperCollider, run the script command Quark.gui using cmd + enter, select Bjorklund from the list, save, and recompile the class library.

2.1 Rhythmic Sequencing

While generating a rhythm for one instrument is novel, the complexity emerges when overlaying multiple timbres with altered patterns. The initial implementation focused on four timbres; kick, snare, closed hi-hat and open hi-hat. The newest version, however, makes use of waveguide mesh physical modeling synthesis and phase modulation synthesis to create more dynamic sounds, blending the notion of a drum machine with a melodic sequencer. Each timbre has a separate rhythm sequence associated with varying fill and pattern lengths up to 64 steps. All sequences are quantized and run at a fixed tempo, which can be updated at any time. The interface required to update these patterns defines the interaction. In this implementation, a pattern is generated for the queue once the user selects a sequence length and fill amount. Immediately after the trigger button is pressed, the new pattern is queued to begin on the onset of the next sixteenth note. This prevents latency issues and timing errors.

Figure 2: Screenshot from Wills Glasspiegel's mini-documentary on Footwork for National Public Radio"

This approach is a variant on the frenetic nature of footwork's odd-time rhythms. Producers like DJ Rashad and DJ Spinn of the Teklife crew used modern tools like Ableton Live and Akai MPCs to sample, produce, and perform their music [1]. The trigger based approach and polyrhythmic structures are reminiscent of their signature style. Whereas techno often uses E(4,16) against E(5,15), footwork is more eclectic. The foundation of many footwork rhythms include E(6,16) or E(15,40) constantly shifting and re-triggering the kick drum and bass synths.

2.2 Pitch Sequencing

Along with the rhythmic sequencer, each timbre contains a corresponding pitch sequencer. Like the rhythmic sequences, the pitches only update upon pressing the trigger button so manipulation can be done unobtrusively. The 16 step multi-slider is mapped to an odd tuning system for microtonal melodies to promote gestural or painterly exploration. The pitch scale is generated by the pixel height, currently it is designed to be 110 pixels scaled between 50 and 13000 Hz mapped logarithmically. "Our perceptions are logarithmic, not arithmetic, and that is important. Rhythm has it's own field of perception and between eight and sixteen seconds there is a transition between our perception of rhythm and form." [3] When a rhythmic sequence length is a multiple of 16 steps, the pitch sequence repeats in phase. When the rhythmic sequence is say 17 or 31 steps long, the pitch sequence begins to phase, creating a longer and more complex melodic phrase. The pitches are mapped precariously within each timbre's synth definition with portions of the construction, such as the pitch slightly modulated with mouse movements on the user's trackpad. This abstraction forces confusion and relinquishes control. An intuitive and inevitable approach to interaction is to swipe through the sliders with one's mouse, painting a pitch sequence.

2.3 Timbre generation

The four timbres used in this iteration are synthesized with SuperCollider. The initial implementation used modified examples for kick, snare and hi-hat sounding synth voices. These sounds were intentionally synthesized so that they could be extended and further developed. The new timbres were carefully constructed for stereo-phase scope visualizations while covering a variety of dynamic percussive sounds. There are no audio effects such as reverb or delay added, the synthesis is entirely dry so that the generative and visualized elements remain transparent. The timbres themselves are heavily modified version of SuperCollider demonstrations [6] for the built in unit generators. They are generally based on the hexagonal waveguide mesh physical model for drum membranes and a phase modulation oscillator pair unit generator, which is akin to frequency modulation synthesis. The envelopes are short and percussive for a fast-paced pointillistic aesthetic. The carrier pitches for each synth definition are slightly detuned between the right and left channels to create binaural beating effects, which register on the stereo-phase scope. Mouse movements in the X and Y plane slightly modulate the amount of detuning to create slight variations in the pitch and thus the visuals. Lastly, the stereo image is panned at low frequency rates using various waveforms including noise, triangular waves, and phase modulation.

3. IMPLEMENTATION: VISUAL AND HAPTIC RESPONSE

3.1 Graphical Interface

Figure 3 displays a screenshot of the graphical interface. Each of the four timbres in the graphical interface contains a slider, a scrollable number box, a trigger button, and a pitch multi-slider. At the bottom lies a tempo number box and its corresponding trigger. The slider and its label represent the fill amount within the sequence. The number box displays the sequence length. The fill amount cannot be longer than the sequence length and the UI ensures this with validation. The fill amount will also update to estimate the rhythmic density if the sequence length is expanded. For example, if the fill length is 9 in a sequence of length 16, the fill will update to 18 when the sequence length is updated to 32. This becomes useful when changing to less common sequence lengths. The *Trigger* button updates the calculated sequence and pushes it into the queue. Multiple patterns can be triggered at the same time.

3.2 MIDI Controller Interface

The GUI can be controlled externally with a USB MIDI controller. Currently, the MIDI CC configurations are hard-coded to match Program 1 and 2 of an Akai LPD8 Laptop Pad Controller as shown in Figure 4. Note that Pad 1 begins on the bottom left and Knob 1 begins at the top left. MIDI CC knobs K1 and K2 control the slider and number box respectively for the first timbre. The top left pad corresponds to its trigger button. Knobs K3, K4 and pad 7 map to the second timbre; K5, K6 and pad 1 to the second; and K7, K8 and pad 3 to the third. On the next program, Knob K1 maps to the tempo and any of the 8 pads map to the tempo trigger.

The sequence length is set to a maximum of 64 steps because of resolution issues with the MIDI CC knobs. Ideally this length would extend beyond 64 steps. Many of the existing interfaces, especially the afformentioned hardware units, are limited to 8 or 32 steps. Expanding this length to 128 steps becomes problematic because the user cannot easily specify the fill amount and step length at that resolution. At a 128 step resolution, it becomes frustrating to align the CC knob to specific fill values, often landing 1 value above or below the desired number.

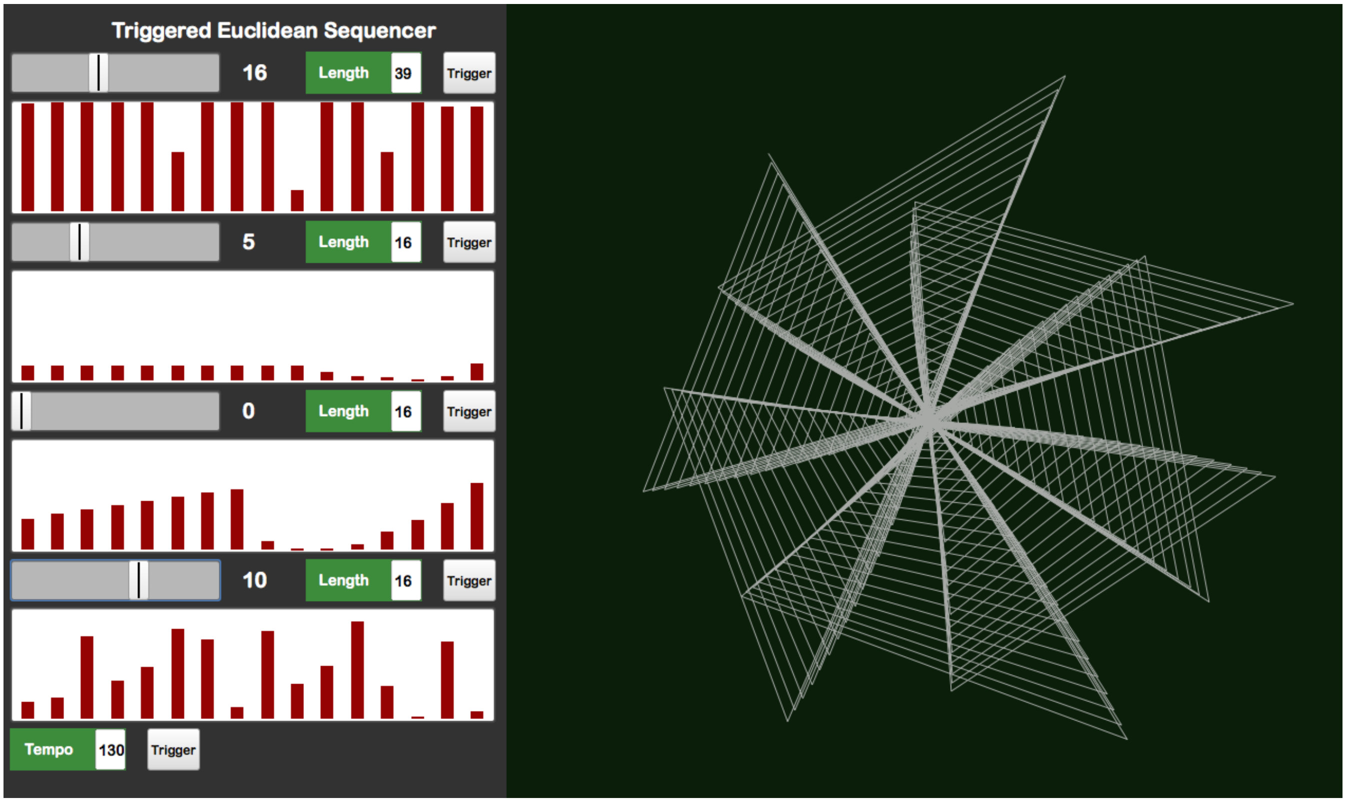

Figure 3: Euclidean Sequencer: Graphical Interface

Figure 4: Akai LPD8 Midi Controller [4]

3.3 Visualization

Rather than focusing on a straightforward visualization of the euclidean sequence, the instrument provides a modified stereo-phase scope graphic generated using the 3D canvas feature in SuperCollider. The animation slowly rotates around its three axes to display variations on the binaural images produced by the synth definitions. Various pitches configurations, mouse modulations, and combinations of timbres produce high-speed bursts of 3D meshes. The image evokes the epileptic visuals associated with the Raster-Noton artists' live performances, aesthetically accentuating the microscopic elements of the system [10]. The visual feedback acts as a source of primary feedback motivating the user to create subtle changes or drastic changes in their sequence as if painting with sound. A secondary visual mode for projection splits the combined scope into four distinct graphics corresponding to each timbre. The visual doubles as a video projection for a performance environment, providing transparency for the audience by communicating the performer's actions. The scope itself also explains the sometimes jarring high frequency sounds designed within, which often create some of the most stunning visuals.

4. INTERACTION

Figure 4 describes the interaction model for a digital musical instrument. The primary feedback [9] of the system, is the graphical interface on the laptop screen and generated visualization. The GUI provides the user with the sequence length, fill amount values, and a set of pitches for each voice. The sliders and number boxes update in real-time with the

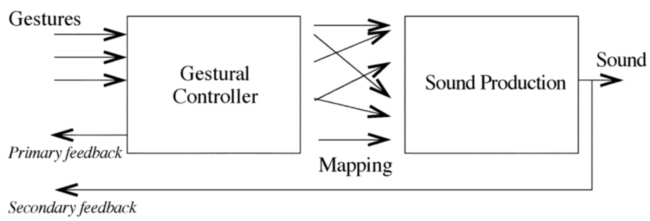

Figure 5: Symbolic Representation of a DMI [9]

MIDI controller's knob changes. The fill amount reflects updates to the sequence length so that fill value cannot exceed that of the sequence length. The pitch multi-slider is controlled by the user's laptop trackpad. When the app initially loads, all the pitches are set at the halfway point. Intuitively, the use swipes through to generate a spline-like shape. After some investigation, more specified sequences are selected for improvisation. Certain timbres work well with different pitch settings. For example, voices one and two have effective kick sounds on the lower registers and intense high frequency visualizations generated in the upper register. A sweep of pitches generates an arpeggiated melody for the resonant construction behind voice three while highly varied pitches create rich percussive sounds for voice four, the waveguide mesh model.

Secondary feedback informs the users decision regarding form. The secondary feedback loop constitutes the polyrhythms and phasing melodic sequences generated by the tool [9]. The user's decision to update the sequence is based on the current pattern and timing. Users decide, for example, when to remove a voice, add density to the rhythm, or re-trigger the melody. The entire system relies on chance, where interrupting the current state, even by simply re-triggering the same pattern, will result in a new combination. Captivating polyrhythms emerge from the use of prime numbers when sequences are not multiples of one another. Phasing microtonal arpeggiations emerge when the sequence length is not a multiple of 16. This creates more complex woven patterns that stray from typical repetitions in rhythm-driven electronic music. This secondary feedback loop is what informs the musician when to disrupt the current state only after locking into maybe one of the rhythmic subdivisions. This mathematical response to update the sequence from the current pattern is processed in both the primary and secondary feedback steps.

The visual response, on the other hand, is another way to receive secondary feedback from the system. Russell Haswell articulated this phenomenon in his 2012 piece 'Oscillographics: the search for unusual images on a Stereo-Phase Scope', featured the on Ràdio Web MACBA series, *Composing with Process.* Decisions regarding density and relative repetitiveness are highly dependent on the visuals in which the performer wants to find. Dense and chaotic rhythms will create a tangle of lines and angles while minimal or repetitive rhythms can generate complex spiral patterns that can be slightly modulated with mouse movement.

5. CLASSIFICATION AND EVALUATION

According to Robert Rowe's typology for interactive music systems, the triggered euclidean sequencer is performance-driven, generative, and sequenced under the instrument paradigm [5]. The environment for which this instrument is intended and its modes of interaction support the notion of a performance-driven classification. The pitched component covers the sequenced classification and the euclidean rhythms cover the generative aspect.

The extension of pitch, tempo, and synthesis design has significantly extended the capabilities of the instrument. Like the Mutable Instrument's Yarns module [2], the system now performs Euclidean arpeggiation, however under a different guise. The fixed sequence length enables 4/4 repetitions to become more traditionally melodic while odd time signatures do not. One evaluator recognized the influence of footwork music on the instrument's design calling it a "footwork generator".

Ideally, the program would run on a touch screen device like an iPad to swipe across the pitch sequencer. This would invite the painterly gesture more so than the computer trackpad. However, the trackpad or mouse interface is efficient for selecting more specific pitches. An indicator to display the pitch index in the sequence and a visual display of the euclidean sequence would allow the musician to have a full understanding of sequence parameters. Because portions of the UI generation are encapsulated in a function, the program's architecture made developing these tasks nontrivial. There are some internal structures that the interface does not exploit, like the ability to easily shuffle or randomize the pitch sequences and multichannel output for recording or triggering alternate visuals.

Further development of MIDI controls would also improve this tool. Most importantly is pulse (trigger or gate) and pitch output. To appeal to expert users, sending MIDI out data would allow expert users to sequence external hardware synthesizers. Midi, however, becomes problematic because the pitch sequences do not follow the typical standard of A440 in equal temperament. OSC or control voltages are another option. Basic functionality for saving and recalling sequence presets is helpful for solidifying compositions. The ability to trigger and accept program changes to alternate samples, synth patches, or visuals would be useful to more advanced users. Lastly, mapping tempo to vary envelope and even tuning parameters would articulate the idea of melding melody, rhythm, and form.

6. CONCLUSION

The triggered euclidean sequencer provides a captivating interface to create pitched polyrhythms. The combination of familiar gestures with it's reliability allows this tool to be used to create a rich and immersive live performance. Along with some of the hidden features, it is designed for stereo or multichannel setups, multitrack recording, and multiple projections. As an example of interaction, the tool is successful in that it provides primary and secondary feedback for the user to direct movement and form. The timbres blend the distinction between tonal and percussive music and exploit discretized approaches to composition. The next approach for this approach to generative music may be to replace the discretized model by a stochastic one to explore density and grains rather than micro pitched phasing rhythms.

7. REFERENCES

[1] W. Glasspiegel. Chicago's footwork music and dance get a transatlantic lift. http://www.npr.org/blogs/therecord/2010/12/14/131834348/chicago-s-footwork-music-and-dance-get-a-transatlantic-lift.

[2] M. Instruments. Eurorack modules. http://mutable-instruments.net/modules.

[3] R. Maconie, editor. *Stockhausen On Music*. Marion Boyars Publishers Ltd, London, 1989.

[4] A. Pro. Media resources. http://www.akaipro.com/company/media-resources.

[5] R. Rowe. *Interactive Music Systems: Machine Listening and Composing*. MIT Press, 1993.

[6] D. C. Scott Wilson and N. Collins, editors. *The SuperCollider Book*. The MIT Press, Cambridge, MA, 2011.

[7] R. Technology. Stoicheia. http://www.rebeltech.org/modules/stoicheia/.

[8] G. T. Toussaint. The euclidean algorithm generates traditional musical rhythms. *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, pages pp. 47–56, July 31 to August 3 2005.

[9] M. M. Wanderley and P. Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4):632–644, 2004.

[10] R. Young. Raster-noton: Impulse to line. http://www.bbc.co.uk/radio3/cutandsplice/rasternoton.shtml, 2005.

*All links accessed April 30, 2015