12-2018

Audio to Architecture: House Music as a Form Generator

Polina Timchenko

AUDIO TO ARCHITECTURE:
HOUSE MUSIC AS A FORM GENERATOR

By Polina Timchenko
Fay Jones School of Architecture and Design
University of Arkansas
Architecture
11.29.2018

Committee:
Chair: Jeffrey Quantz (Architecture)
Members: Jeff Shannon (Architecture)
Ken McCown (Landscape Architecture)

TABLE OF CONTENTS

PROJECT SUMMARY

Contemporary music undergoes similar process of creation to that of the design process through computation and variation. House music as a representation of contemporary culture has a layered structure that allows specific characteristics to identify it as house music. Song components can vary and mix in different orders that form new dynamic compositions. I am going to explore the idea that every house music component can be translated into geometry with the use of parametric design techniques.

PROJECT DEVELOPMENT PLAN

Week 1:
theory discussion and data collection;
answering what kind of data songs have;
getting familiar with Ableton Live 10;
getting familiar with the song “looking back”;
extracting separate tracks and exploring their parameters;
converting 1 track to mp3 format and importing it in grasshopper

Week 2:
converting the rest of the tracks to mp3 and importing them into grasshopper;
offsetting each track and establishing grid;
studying physical properties of the sound, looking for parameters and definitions;

Week 3–4:
establishing grid based on song parameters;
studying data trees;
watching grasshopper tutorials;
looking for another song;
separating song into segments;
studying a smaller segment;
aligning tracks with the grid;
extracting numeric data out of each segment;

Week 5–6:
cleaning up the script;
assigning geometry to a single cell;
layering geometry;
studying one segment at a time;

Week 7–8:
looking at tracks as a whole;
getting familiar with a new song;
preparing the song for input;
displaying full track on grid with geometry;
displaying all tracks;
offsetting tracksa

Week 9–10:
review with outside critics;
presentation preparation;
addressing feedback;
re–arranging geometry to linear array;
changing the song to “slippery when wet”

Week 11–12:
adjusting geometry;
displaying new song;
making the script automatic;
documenting

Week 13–14:
minor adjustments;
documentation

Week 15:
writing,
editing,
presentation set–up,
defense

INTRODUCTION

The project started out from my love and passion to electronic music. I've focused on this particular type of music because it is made in the most contemporary way and doesn't require any instruments to create a composition, the only tool you would need is a computer with sound editing software, such as Ableton Live, which was used for this project. Technologies are inevitable parts of the design process that went beyond the simple definition of a technique. We are not required to be proficient in a new field in order to notice technologies and creative process applicable directly to our field. [1] Music creation and design share similar process of development through layers and variation.

House music is one of the genres of electronic dance music. It first appeared in clubs of Chicago in the early 80's and made its way to contemporary music scene gaining its popularity not only in clubs and festivals but also in everyday life through movies, shows, commercials and radio. [2] For several years I've been observing house music scene from underground clubs to major music events and I've decided to learn a little more about its creation process and structure.

Creating a musical composition was not my intention, I was interested in how the music could be represented and be viewed as a form. Is it possible to extract form out of music? How will it look like? What are the steps to extract form out of sound?

In search of the answers I turned to computation strategies, which is the use of computer to process information through an algorithm. An algorithm is a particular set of instructions that must be written in a language that computer can understand. Algorithmic thinking requires understanding the code with ability to modify it and control the output. [3]

Music has various parameters that could be used in form generating process. Each song has beats, measures, length, sound waves with their properties: frequencies, amplitudes etc. songs are created in layers of different tracks. Each track usually carries one sound, such as drum track – only carries drums. Layers are used for the sake of clarity and ability to manipulate each bit of a song separately. This process is very similar to design process where most of the components are managed in layers and eventually become one complete object. [4]

With parametric design tools it became possible to control every step of the design process, test countless variations and adjust the process at any point. Music has multiple parameters that can be used in the process to generate geometry.[5]

To analyze a song at its core I needed to dissect a composition down to tracks which would give me more precision in extracting parameters of each track separately. After studying house music characteristics I organized tracks based on their significance in the song, which defined the order in which geometry will be represented.[6]

I've worked directly with Lucas Feazel – a DJ who helped me finding a program to use, navigate through that program and get a file that contains song broken down to tracks.

1. Goulthorpe, Mark, Amanda Reeser, Ashley Schafer, and Alayna Fraser. "Precise Indeterminacy Three Projects by DECOI and an Interview with Mark Goulthorpe." PRAXIS: Journal of Writing Building, no. 6 (2004), 30–31.
2. "A Brief History of House Music." Complex.com.
3. Peters, Brady. "Computation Works: The Building of Algorithmic Thought." Architectural Design 83, no. 2 (2013): 8–15. doi:10.1002/ad.1545.
4. Hydlide. "Basic Elements: House Music." Reason Experts. 10.2016
5. Carpo, Mario. "Big Data and the End of History." International Journal for Digital Art History. Accessed November 29, 2018. http://nbn-resolving.de/urn:nbn:de:bsz:16-dah-499134, 23
6. Hydlide.

Lucas was also the author of every song that I've used in this project.[7] [8] [9] After extracting tracks from sound-editing software I've started working with algorithmic modeling program – Grasshopper, which was my main tool for the rest of the project.

First steps in Grasshopper required hours of video tutorials and related articles from various websites and forums[10], and a direct supervision of my mentor and committee chair – Jeffrey Quantz. After that I was able to import tracks into the software and have my first graphic representation of a song as two-dimensional wave-form. I knew it was possible to extract other parameters from the song and from there I've started writing my algorithm and experimenting at first with segments of the tracks, then with each track separately and eventually used all of them to generate geometry. In this process I've continued working with Lucas on song structure as I've noticed that the file that I was manipulating was missing significant qualities that would make it a house song. He created a new composition that I was able to test through my algorithm and that satisfied my genre criteria.

The project is an opportunity to show that music creation and form making could be very similar processes and can depend on each other. I was able to set a sequence that can be used in design process and eventually transform music into usable space. Further development of this algorithm could work for both musicians and designers and affect both music and form. It is a new tool for us to use and a new way of seeing the music.

7. Feazel, Lucas. "Looking Back". 2018 MP3
8. Feazel, Lucas. "Remedy" Mike Mago. Remix. 2018 MP3
9. Feazel, Lucas. "Slippery When Wet". 2018 MP3
10."Grasshopper." Grasshopper. Accessed November 29, 2018.

BACKGROUND

House Music: Brief History and Structure

House music is the genre of electronic music that was conceived in Chicago in 1980's in underground dance clubs. Today house music has a very diverse audience all over the world and takes on different subgenres depending on the regions where it is produced. For example, deep house is common in Chicago, Detroit, New York, San Francisco areas, it usually stays within the range of 120–135 beats per minute (bpm) and sometimes uses vocals, while tropical house subgenre is more common in Miami and Ibiza, and it has slower tempo, usually sounds more melodic and more likely to use vocals than deep house. Each song has measure count, where typical house song would have 4 beats per measure. Songs generally last about 3 minutes and usually contain nearly 100 measures.

Although house music has been around for almost 40 years, it is only now becoming widely accepted as we hear it beyond the club scene on the radio, music festivals, commercials, as a background in public spaces and see it in our suggestions for popular online music channels. It is involuntarily becoming a soundtrack to our lives; it's the music of our time.

Every house composition has a structure that consists of underlying beats:

– bass drums set up the rhythm of the composition, most often bass drum plays 1/4th of a note;

– irregularities to the bass drums that are also knows as syncopation are created with claps and snares, these elements set the main mood of the composition — the more frequent they are the more aggressive the composition gets.

The rest of the drum related elements are various percussions, hi–hats (hats) and shakers, they are all there in order to make the drum elements more diverse and develop the composition in the desired manner. Other popular instrument tracks that are used to color the composition could be string instruments such as violin and woodwind instruments like saxophone.[11]

Effects (fx) — group of sound rendering elements such as reverb or delay to speed up, repeat or slow down parts of a track, they add another layer of complexity to the song.

Some songs can also contain vocals to add another layer of depth to the composition. House tracks are usually very concise with the message: it could be just one word or one line that highlights the mood of the song. Vocal samples can be distorted and repeated several times over the song.[12]

House music rarely records entire composition from live instruments since most of the necessary sounds can be found online or in software library for personal use. There are millions of opportunities to mix those sounds in one of a kind pattern and get a new sound every time. Many artists take advantage of sounds that catch their attention either in songs of other genres or house songs by other artists and combine them with their own sound samples. Songs that borrow primary elements from other compositions are called remixes.[13]

Electronic music earned its name through the process of its creation, which mainly involves sound editing software. Music is created from combination of sound samples that come from different sources or can be custom made. Those samples are manipulated into tracks, rendered with additional effects and then combined into a song.

11. Complex
12. Hydlide
13. Hydlide

Project Tools

Music editing software:

– AbletonLive 10 – music making and editing software, helped reading and manipulating tracks on initial stages; gave ability to extract each track separately in .wav format

– Wavosaur – helped read tracks and transforms them from .wav audio files into .txt text format that can be read through any word processing software. Wavosaur helped converting tracks to .mp3 format, which is widely acceptable audio format and was necessary for other programs to understand each track.

3-D modeling:

– Rhinoceros 5.0 (Rhino) – necessary to read algorithmic outputs and run other algorithmic modeling plug-ins

– Algorithmic modeling

– Grasshopper – was used to write main algorithm[14]

– Mosquito – grasshopper plug-in – used to bring in .mp3 tracks and display them in Rhino[15]

Literature Review

Algorithms Aided Design by Arturo Tedeschi was one of the key manuals on algorithmic work flow and process, describes basics of what algorithms are and how they can help the design process. This manual gives tips on the software and also teaches how to be in control of our designs while using computation.[16] From Control to Design by Tomoko Sakamoto had multiple examples of buildings and their elements generated by parametric design tools.[17] The rest of the theory and examples of parametric design came from journals Praxis (issue 6)[18], Architectural Design[19] and International Journal for Digital Art History[20]. These pieces were necessary to see examples of how algorithmic design works and understand some of its endless possibilities. Each of the sources contains theory of what computation is and how we as architects can relate to technologies.

Another part of the learning process was learning each program, mainly through tutorials. Basic tutorials were found on Lynda.com, some examples are “Grasshopper essential training”[21] and “Learning Grasshopper”[22]. After these introductory courses I had to be more specific in my questions and search for each necessary component separately. Most of the answers were found through Grasshopper forums on grasshopper3d.com[23].

Music theory comes from articles and blogs that described physics behind the music and history behind house music. Sidney Wood at SWPhonetics describes physics behind the soundwaves. “Understanding Waveforms” was helpful during the search of parameter and explained how waveform diagrams are generated and what they mean.[24] It was also helpful to look at the process behind song’s creation, where songs are broken down to tracks and explained step by step. David Felton for Attack Magazine describes the study of several popular songs, including “Can’t Do Without You” by

14. “Grasshopper”
15. Smuts, Carson, Breanna Carlson. “Mosquito.” Studio Smuts. Accessed Nov. 29 2018.
16.Tedeschi, Arturo. Algorithms Aided Design. Potenza, Italy 2017.
17. Sakamoto, Tomoko. From Control to Design: Parametric/Algorithmic Architecture. Barcelona: Actar, 2009.
18. Goulthorpe
19. Peters, Brady. “Computation Works: The Building of Algorithmic Thought.” Architectural Design 83, no. 2 (2013): 8–15. doi:10.1002/ad.1545.
20. Carpo. “Big Data”
21. Reilly, Chris. “Grasshopper Essential Training.” Lynda.com – from LinkedIn. October 26, 2018. Accessed November 29, 2018.
23. Reilly, Chris. “Data Tree Components.” Lynda.com – from LinkedIn. September 26, 2014. Accessed November 29, 2018.
24. Wood, Sidney. “Understanding Waveforms.” SWPhonetics. September 12, 2018. Accessed September 16, 2018.

Caribou.[25] This song was the closest to electronic music and showed how layers in the track work together.

Music has been used in architecture through mathematical relationships – rhythm in a song could be translated into architectural patterns. Charles Jenks in "Architecture Becomes Music" describes historic examples of music and architectural relationships through gothic cathedrals and Greek temples. The article talks about proportions, harmonics and how music has been closely related to architecture throughout the history of civilization.[26]

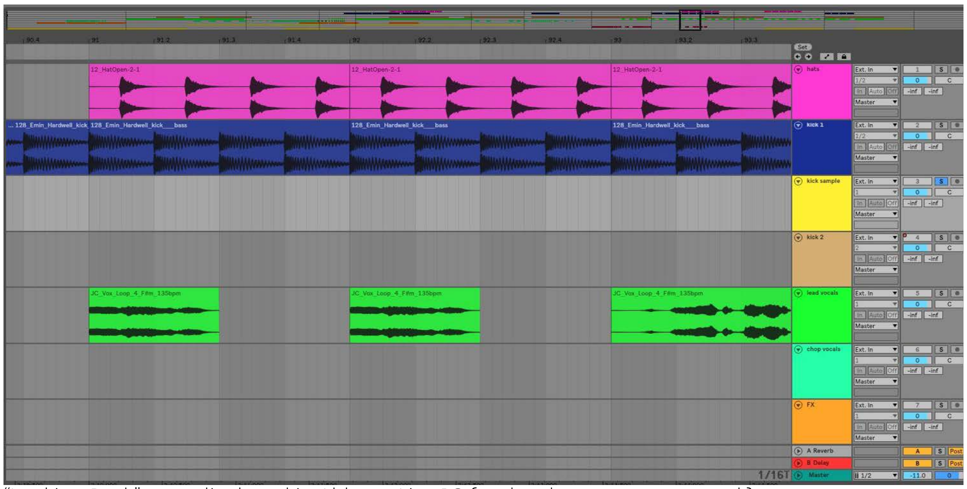
Precedent studies included couple experiments conveyed by John Locke and Colin Morris. They could relate to my project in several areas. John Locke's project became one of the precedents for my experiment since he was transforming soundwaves into a landscape. His experiment focused on fragments of four different songs that later became 3D printed topographies.[27] Locke's work has a direct translation of existing audiowave into form. Colin Morris conveyed his study on popular music and wrote a script that can identify repetitions of words in popular songs. He wasn't aiming for formal representation but he was able to excerpt repetitive data and create visual patterns that would reflect that data.[28] Both of these precedents shaped a direction for my experiment, where I could analyze data and create abstract form that would represent music.

25. Felton, David. "Deconstructed: Caribou – Can't Do Without You." Attack Magazine. 2015.
26. Jencks, Charles. "Architecture Becomes Music." Architectural Review. 6 May, 2013. Accessed December 09, 2017.
27. Locke, John. "Sound." Gracefulspoon.com (web log), 2012. Accessed December 10, 2017.
28. Morris, Colin. "6 Weird Pop Songs Visualized." March 14, 2017. Accessed December 10, 2017.

PROCESS DESCRIPTION

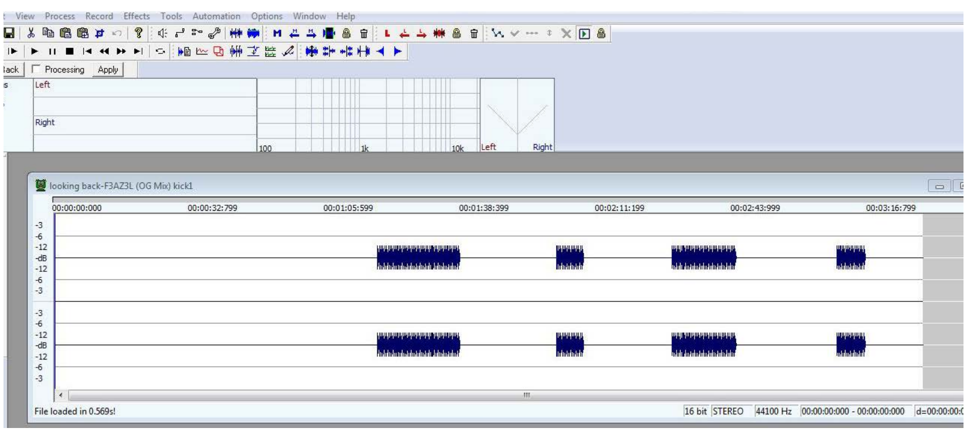
Week 1

The main goal of the week was to extract usable data from audio files. Song "Looking Back" by Lucas Feazel that was provided by him for this project originally contains 9 tracks in Ableton: hi-hats, kicks 1& 2, lead vocals, chop vocals, fx, synth, sax, strings.[29]



"Looking Back" song displayed in Ableton Live 10 (each color represents a track)

I've extracted each of the tracks as .wav (waveform audio format) file in order to manipulate each layer of composition separately. From .wav file using Wavosaur software I was able to extract .txt or text document for each of the 9 tracks.



kick1 track in Wavosaur

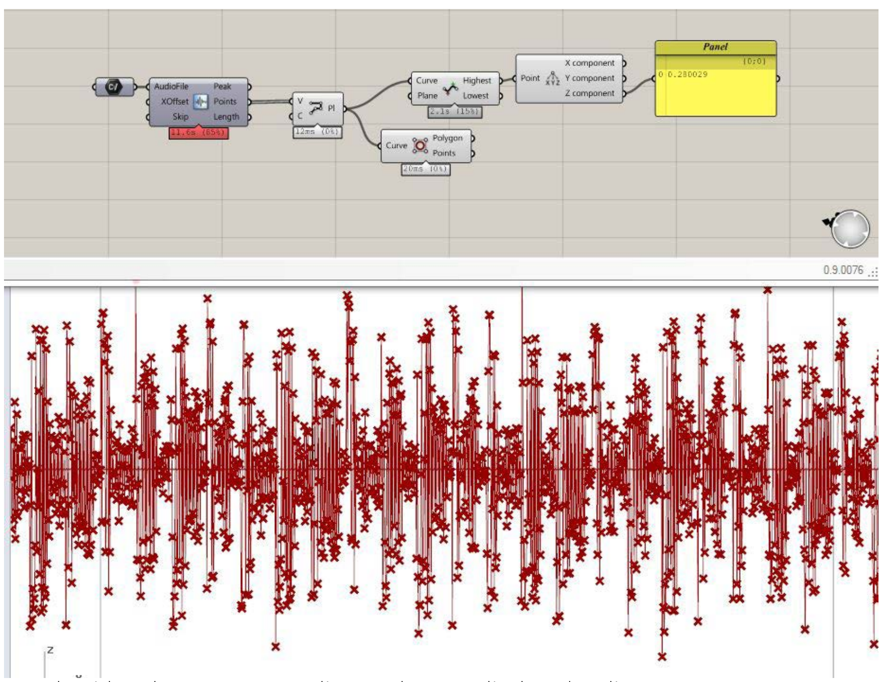
Each of the files produced more than million numeric values (exact number is unknown but definitely larger than 1,048,576 values). I had to use a smaller piece of the track to be able to compare some information. My attempts of using 1/4 of a second of kick1 track – didn't give a needed result either. The piece produced over 100,000 numeric values, where particular pattern wasn't identified besides the white noise that fluctuated between 3 values -0.000031; 0; 0.000031. In addition to that values varied from one attempt of converting audio to text to another.

29. Feazel, "Looking Back"

0.000031	0.000031	-0.006623	-0.006623	-0.011536	-0.011536
0.000397	0.000427	-0.010620	-0.010651	-0.011567	-0.011597
0.000732	0.000732	-0.009644	-0.009613	-0.005219	-0.005219
-0.000855	-0.000885	-0.005402	-0.005432	-0.003906	-0.003876
-0.001160	-0.001160	0.004700	0.004700	-0.004273	-0.004273
0.002472	0.002472	0.010529	0.010529	-0.007202	-0.007202
-0.001190	-0.001190	0.014405	0.014435	-0.010773	-0.010804
-0.005310	-0.005310	0.013886	0.013886	-0.014100	-0.014069
-0.003174	-0.003174	0.010163	0.010163	-0.010132	-0.010132
0.004761	0.004761	0.005829	0.005860	-0.006195	-0.006226
0.007599	0.007569	-0.001801	-0.001801	0.000580	0.000519
0.006958	0.006958	-0.007721	-0.007752	0.006409	0.006409
0.000549	0.000549	-0.009522	-0.009491	0.011109	0.011109

values output for 0.25 sec of kick1 track

Next step was to go back to the files provided with Mosquito plug-in package that included audiowave file, the main requirement for the algorithm to work was an .mp3 file format (smaller file than .wav). I was able to convert earlier exported .wav tracks into .mp3 which produced a wave diagram in Rhino through Mosquito algorithm in Grasshopper.[30] This gave me an opportunity to get the needed information into necessary format to be able to further manipulate extracted data.

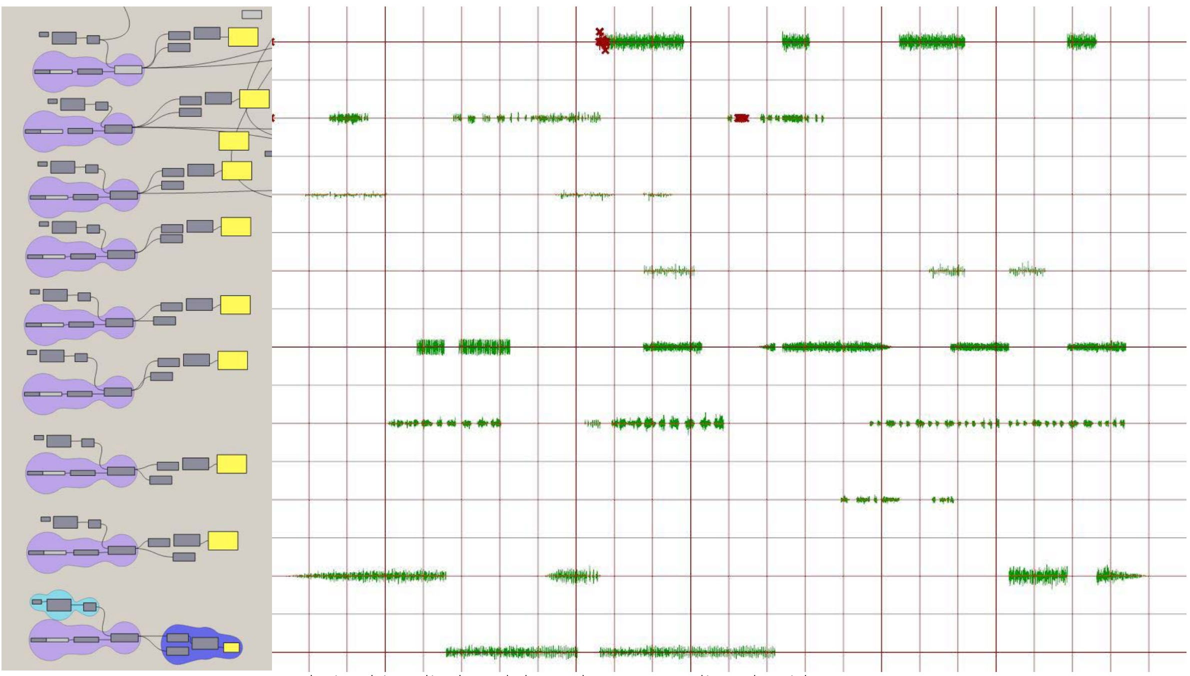


top: algorithm that generates audiowave. bottom: displayed audiowave

30. Smuts. "Mosquito"

Week 2

After plugging each of the nine .mp3 tracks exported from Feazel's "Looking Back" mix into Mosquito I was able to see sound waves for each of them in Rhino. I started experimenting with different data outputs and trying to analyze given sound waves through points of extremes and intersections. All tracks were aligned and divided into 12 measures, which is the measure count in the entire song of 3:35 min. I received even segments that can be studied separately. At this point I know that we can extract data from the songs; however the next step is how to use the data and what exactly is necessary to use the data in architecture.


tracks in Rhino displayed through corresponding algorithm

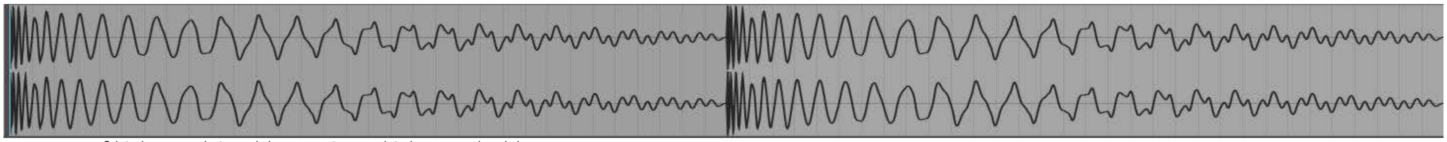
Further investigations led me to studies of sound waves and their structure. Here's what I've rediscovered from "Understanding Waveforms" article by Sidney Wood:

- waveform consists of compression and rarefaction (points of high and low pressure) at the 0 line mark the pressure is "normal", which can happen several times through one wave cycle
- waves have a cycle, that repeats with certain frequency
- high and low points on the waveform are peaks of the amplitude (volume or loudness)
- wavelength is the distance a disturbance travels in one complete wave cycle.[31]

After reading these facts I had to re-establish connections between the composition and the given information. The question that I keep in my head throughout the process is: "what makes house music house?" this time the answer was drums, kick and snare. After listening to "Looking Back" multiple times I realized that in this composition kick tracks don't carry much of significance, meaning that if those tracks are muted – the mix doesn't lose its overall structure, which is unusual for house track.

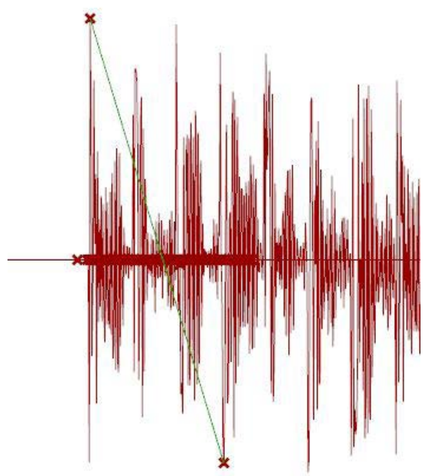
31. Sidney. "Understanding Waveforms"

I've started looking for a track that suits my needs better, and found something that can potentially replace my current sound, which is Feazel's remix of "Remedy" by Mike Mago, however even in this version it was hard to determine solid ground for the composition.[32] At the moment I am staying with "Looking Back" because it has couple of layers of kick drums that I can study, yet I am still in a search for a better composition that would represent house in a better way.



2 counts of kick1 track in Ableton Live10kick 1 track ableton.JPG

From a look at the kick1 track in "Looking Back" I've been able to identify the extremes of each one of 121 regions that divide the track and distance between them. Drums in house music create a foundation for the entire track, without them it will be a set of vocals, effects and instruments that aren't strong enough to keep the composition together on their own. The extreme points on the sound wave describe the wave in compression and rarefaction.[33] This distance measure can establish a grid that will use distance between the extremes as y distance in x; y plane and the cycle length as x distance.



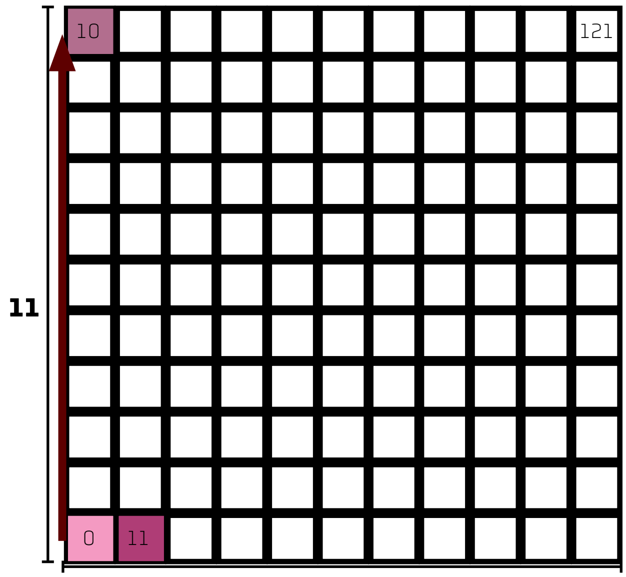
green line representing extremes of one segment of kick1 track

The distance between the extremes from segment to segment equals approximately 0.5". The cycle length is a more challenging task that I am trying to face right now to be able to set up a grid based on kick drums for a foundation of my further architectural investigation.

32. Feazel. "Remedy"
33. Hydlide. "Basic Elements of House Music"

Weeks 3–4

Breaking down song's parameters in more specific way helped me establish parameters for the grid. The song is 3 min 35 sec long, which translates into 215 sec. This parameter was used to define the size of the square field: 215"x215". The square was divided into 122 segments on each side to, which corresponds with the measure count in the song, and, as mentioned before, tracks were divided into 122 segments. The reasons I've based this grid on a square was that each measure in the song is divided into 4 even beats, which suggest a square broken down into smaller squares.



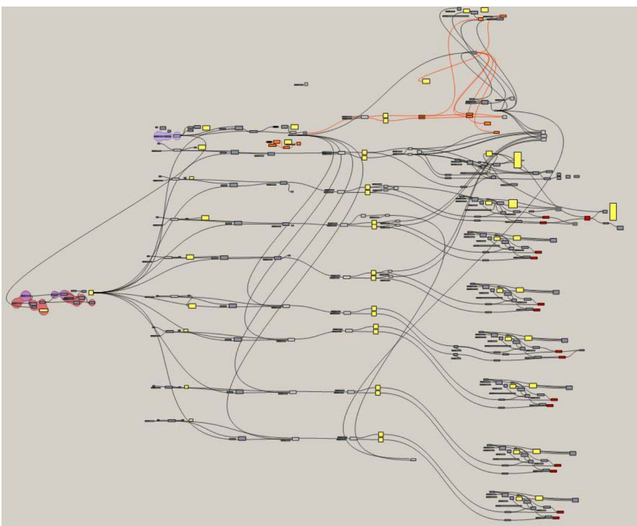
11x11v grid

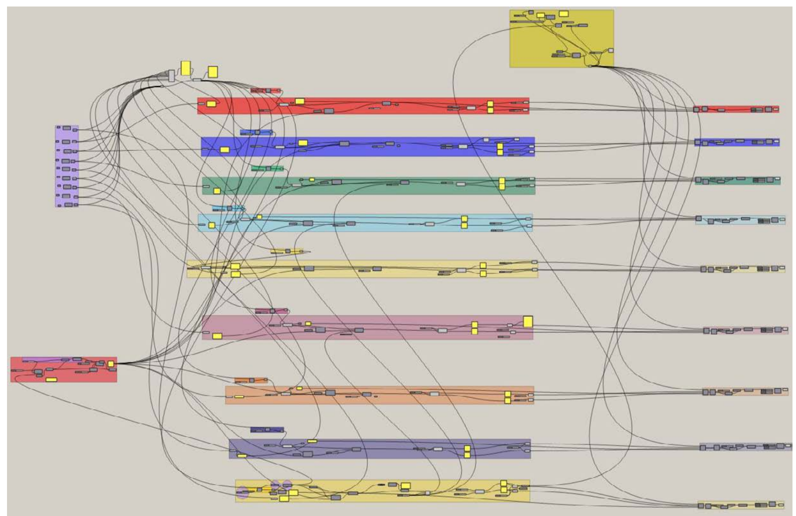
Once the grid was established, data needed to be aligned to it. Speculations on how exactly I need to manage the data took the rout of setting points of max and min amplitude of the wave in each of 122 segments. I returned back to examining the tracks and reset the algorithm to where I am able to register those values where max z>0, and min z<0 on (x;y;z) plane. Algorithm measures distance from baseline of each track, where amplitude=0, to point where z coordinate matches the max or min distance.

To navigate through the grid better, I had to get familiar with the data tree concepts, which brought me to hours of tutorials and explanations on what data tree is and how it works within grasshopper. Short definition is: "Data tree is an ordered collection of lists."[34] Lists contain the data, in my case it is geometry such as points and lines. Data trees allow me to manipulate the data on different levels, for example choose 1 point out of 10 points and work with it. Understanding the data tree helped me manipulate the grid to where I could break it down to rows, columns, cells and points. This gave me control over data excerpts that I can assign to correspond with any given cell.

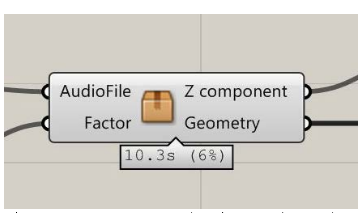
34. Rutten, David. "The Why and How of Data Trees." Grasshopper. Accessed November 28, 2018.

The process took some trial and error and at every step of the way I was running numerous tests to see if each piece functions properly: do the cells correspond to the segments? Is there correct number of segments? Is there correct number of cells? Making sure that each track works correctly and understanding where each component comes from and what it does. There are many ways to resolve each of those questions and write each piece of the program. Sometimes trying different alternatives provide the clearest solutions – clarity achieved by the use of fewer components. That saves time and effort in case something needs to be adjusted in the future. Unfortunately, it is easy to get lost in elements and try to simply get to the goal without thinking about the clarity, which later adds another step of cleaning up and checking the script.

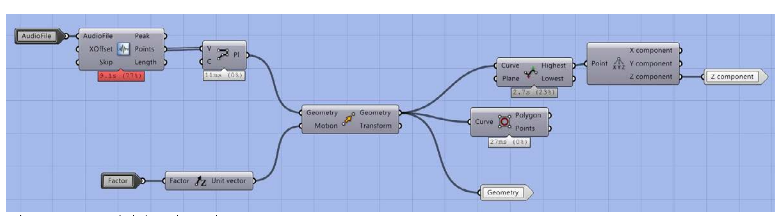

process script.JPG


organized final script.JPG

Through "cleaning process" I've learned how to cluster programmatic elements and reduce the number of "webs" on the screen, especially for the elements that repeat throughout the song, which later helped me resolve several issues that I've encountered. Clarity of the program is just as important as writing it.

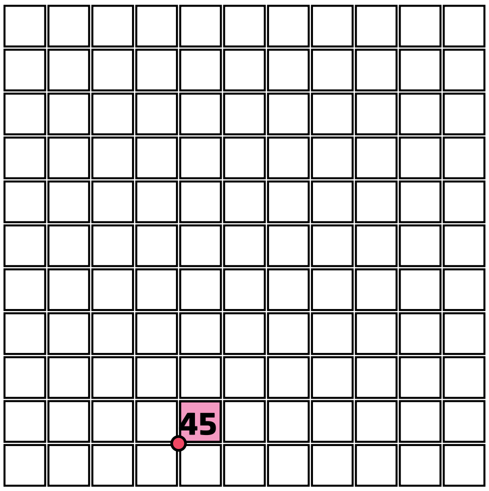


cluster component in the main script


elements within the cluster

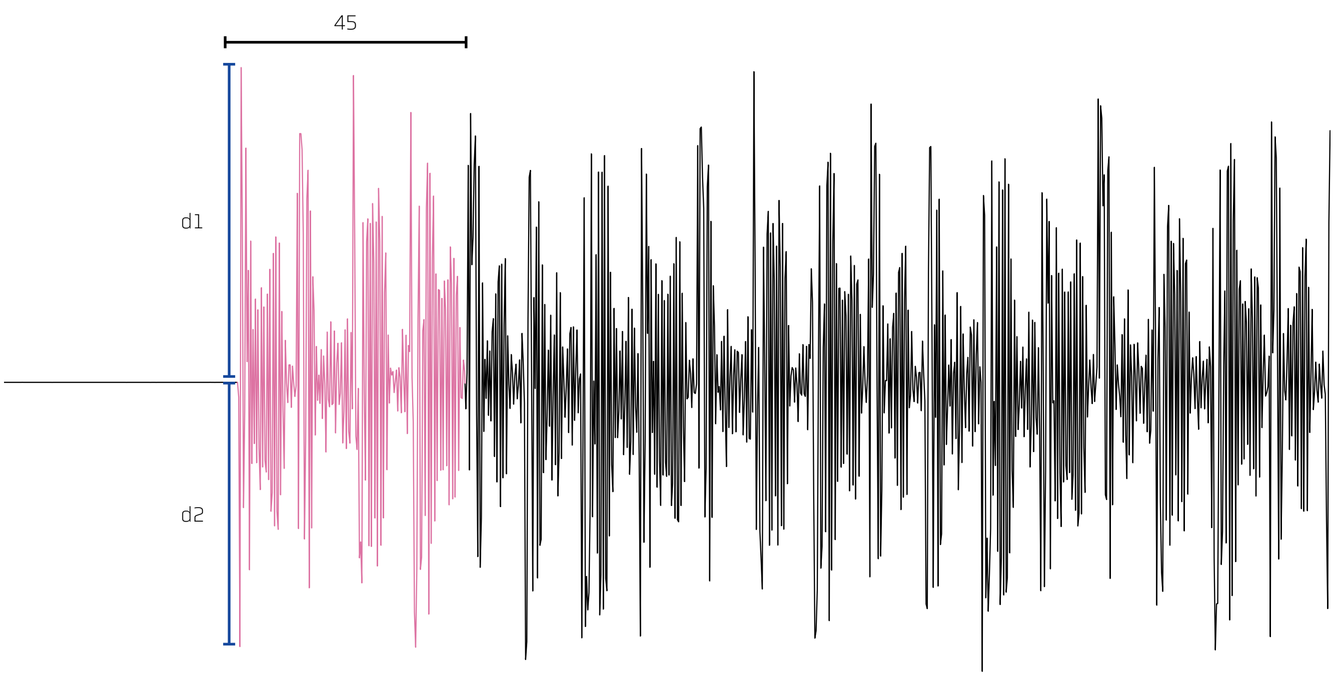
Weeks 5–6

In the next piece of the algorithm grid cells correspond to 122 segments in the song, meaning that when I select a number between 0 and 122 – I select segment of the song and a cell on the grid. To test the theory I had to analyze one of the segments and assign graphic definition. I chose a segment of my first track kick1. It could've been any segment, but in order to see its representation I chose one with defined wave fluctuation.



left: segment #45 on grid
right: orresponding segment on kick1 track

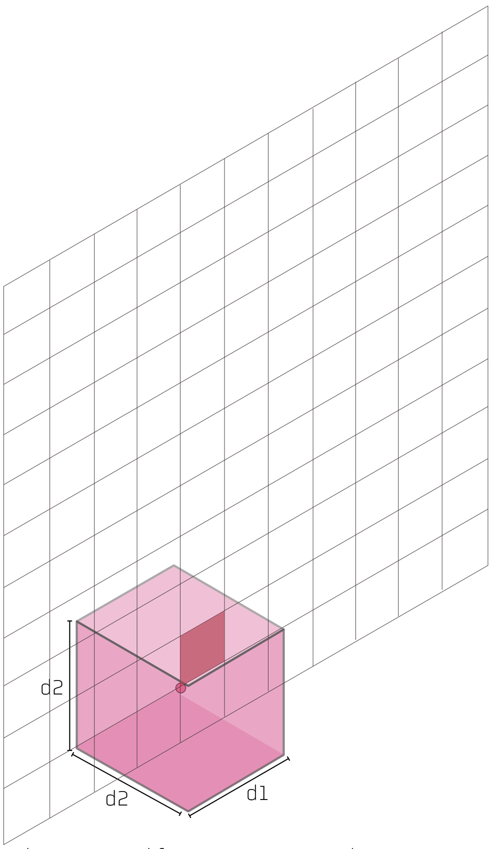
At this segment distance to max and min points were registered and reflected on a simple line length that went through the corresponding grid cell.



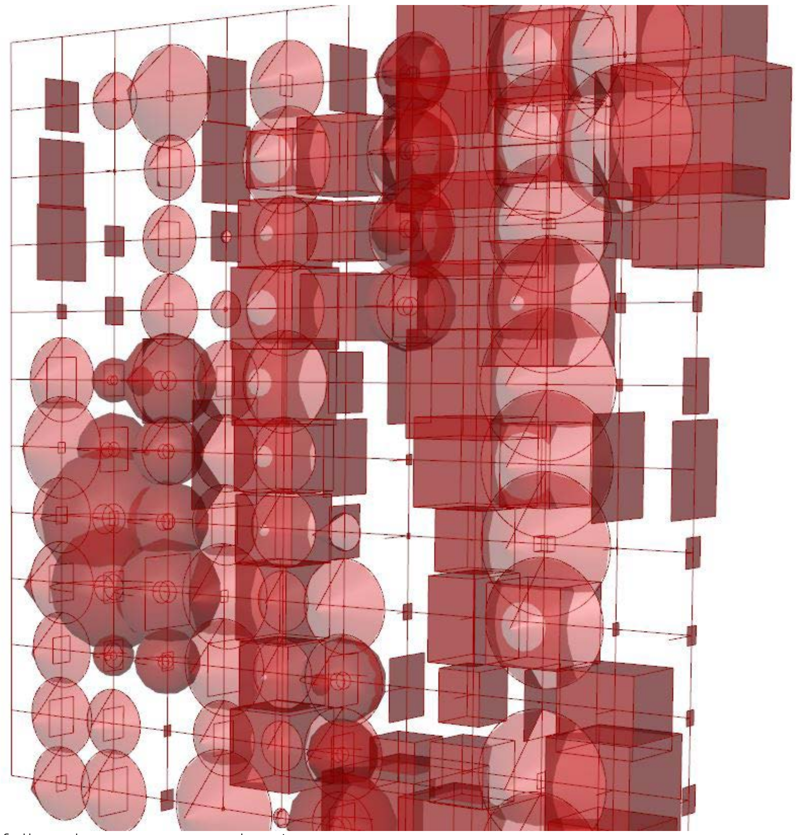
extremes at each segment were identified and measured
they represent the loudest moments of each segment
d1>0;
d2<0

they are used to set the dimensions of each
correcsponding geometry in terms of (x,y,z) values

Then I've experimented with different geometries – squares, circles, cubes, spheres, cylinders and was able to prove that everything works and ready for the next step.



cube generated from segment #45 data


full tracks geometry overlapping

I've identified a segment where majority of the tracks would have values that aren't zero, and it happened to be a segment #45. Different geometry was assigned to each of the tracks. They were reflected at the cell #45 based on the same principle of measuring the amplitude distance in a segment. This result visually represented given parameters successfully, however, geometry was overlapping and only populated 1 cell, as it was assigned.

Another issue that I was facing was that given parameters produced objects with very insignificant differences, which made the object hard to read. To display the difference better I had to add multiplication factor to increase the dimensions of geometry.

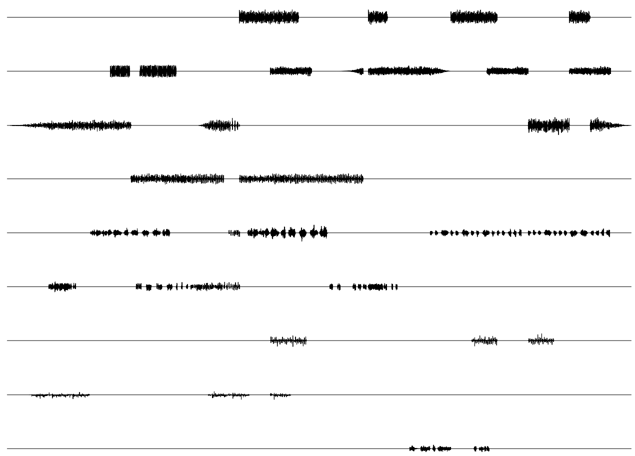
Decision was made to analyze each track fully, using every segment and representing it as geometry. In that way I was able to see full tracks overlap and form an entire song on established grid. Geometry was randomly assigned. Therefore I was supposed to get 122 pieces of geometry per each track, which is 1098 pieces total for 9 tracks per 122 cells. To clarify the image I had to eliminate all pieces that are equal to zero, in that way we can trace the sound wave displayed on the grid.

Experimentation with visual output continued through the change of geometry assigned to tracks. One of the results was displayed through boxes, in order to see all tracks through similar output. This step required an offset of geometries from the grid, which led to a question of specifying offset strategies: which track should come next to another.

To organize tracks I went back to the theory of house music and re-organized the tracks based on their significance in the song.[35]

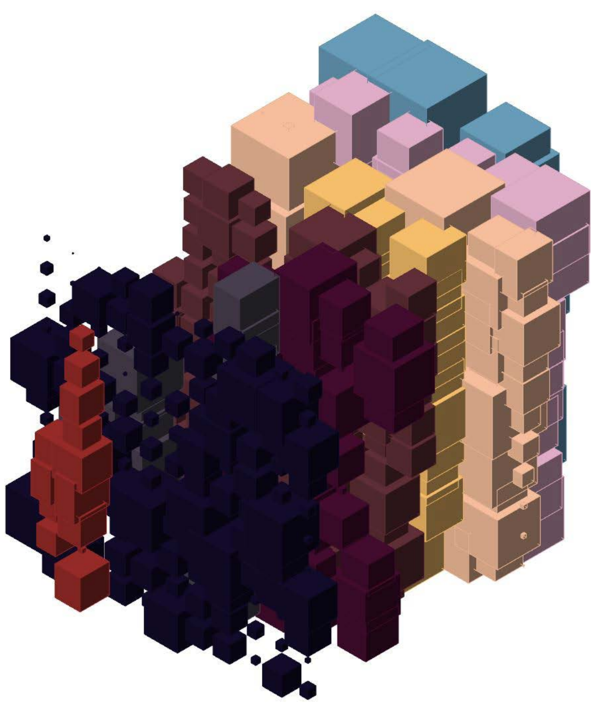
Track's presence in the song was defined by its frequency and wave fluctuation just by listening to the song and visually studying it. Here's the tracks organization rated from most significant to the least:

- kicks1
- kicks2
- strings
- synth
- lead vocals
- chop vocals
- hi-hats
- fx
- sax

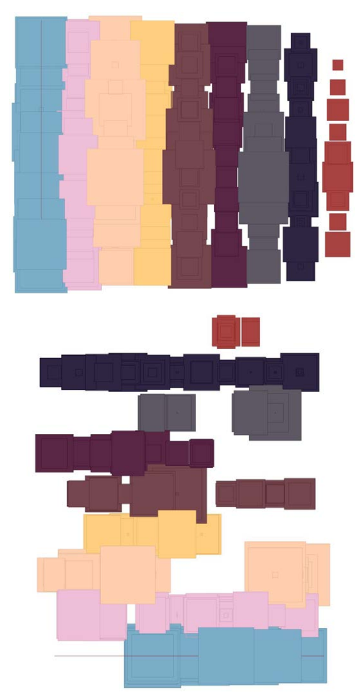


tracks organized

Tracks were offset from each other at equal distances. Colors were added to highlight the difference between each track with RGB values defined by the track's parameters and adjusted for the sake of contrast and aesthetic qualities.



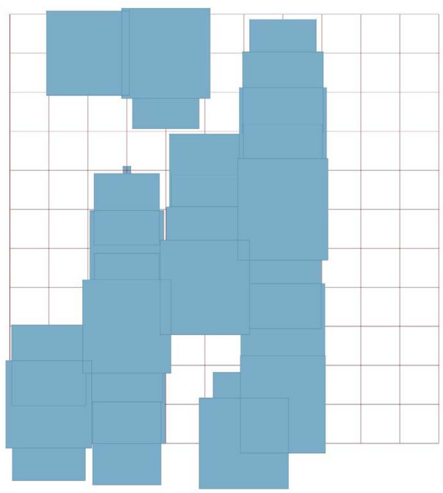
all tracks displayed at once, offset from one another



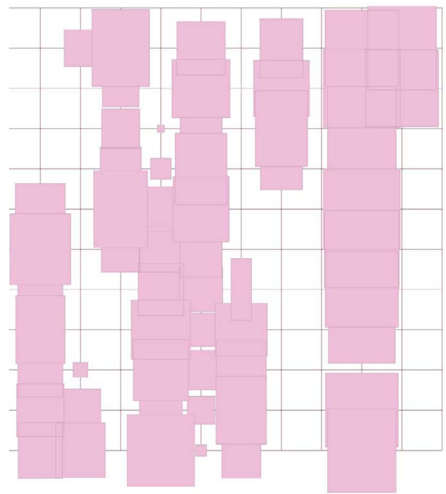
all tracks displayed: side and top views

35. Hydlide. "Basic Elements of House Music"

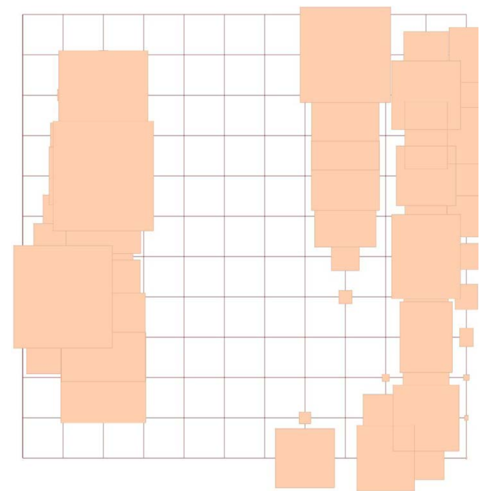
Results were recorded and analyzed. Some tracks produced surprising result, such as fx track, appeared to be much more consistent in its geometry than expected and covered significant amount of the grid. Tracks with the most fluctuations produced the largest geometries.



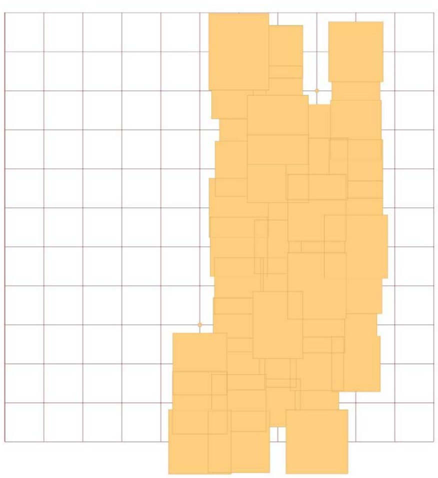
kick1 track



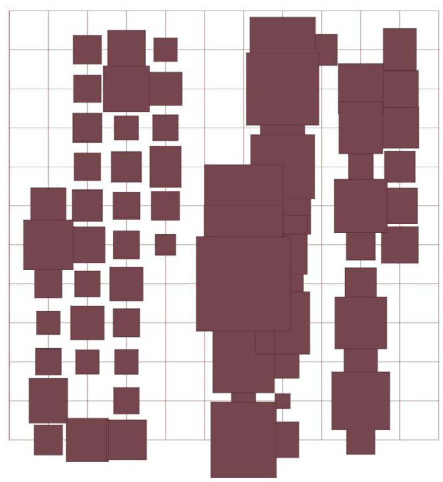
kick 2



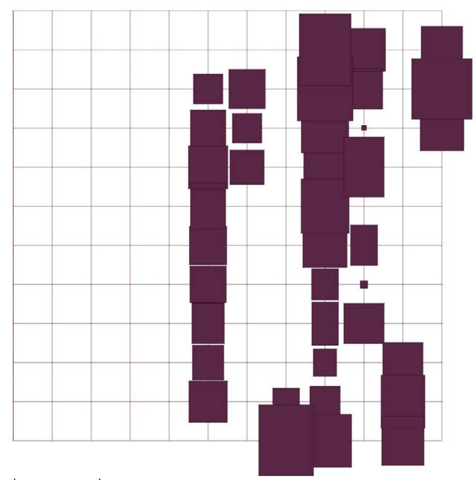
strings



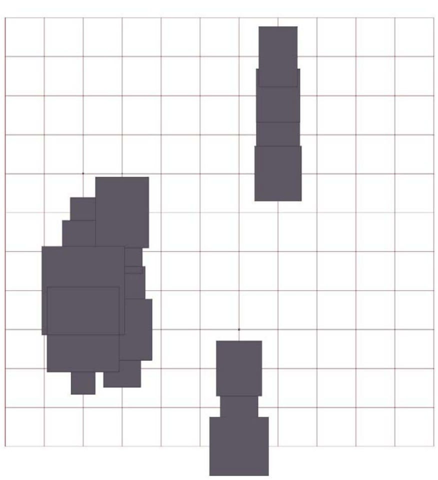
synth



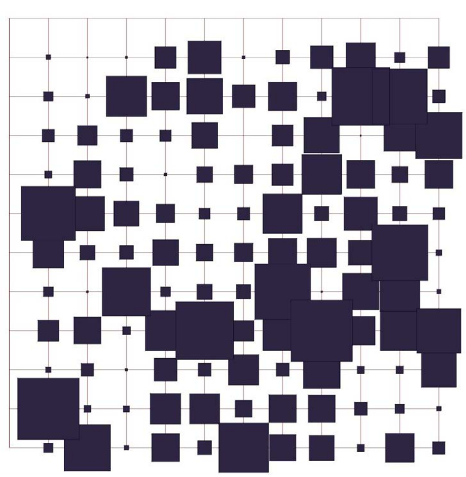
lead vocal



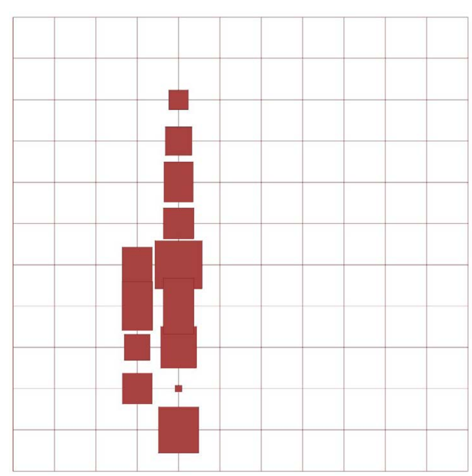
chop vocals



hi hats

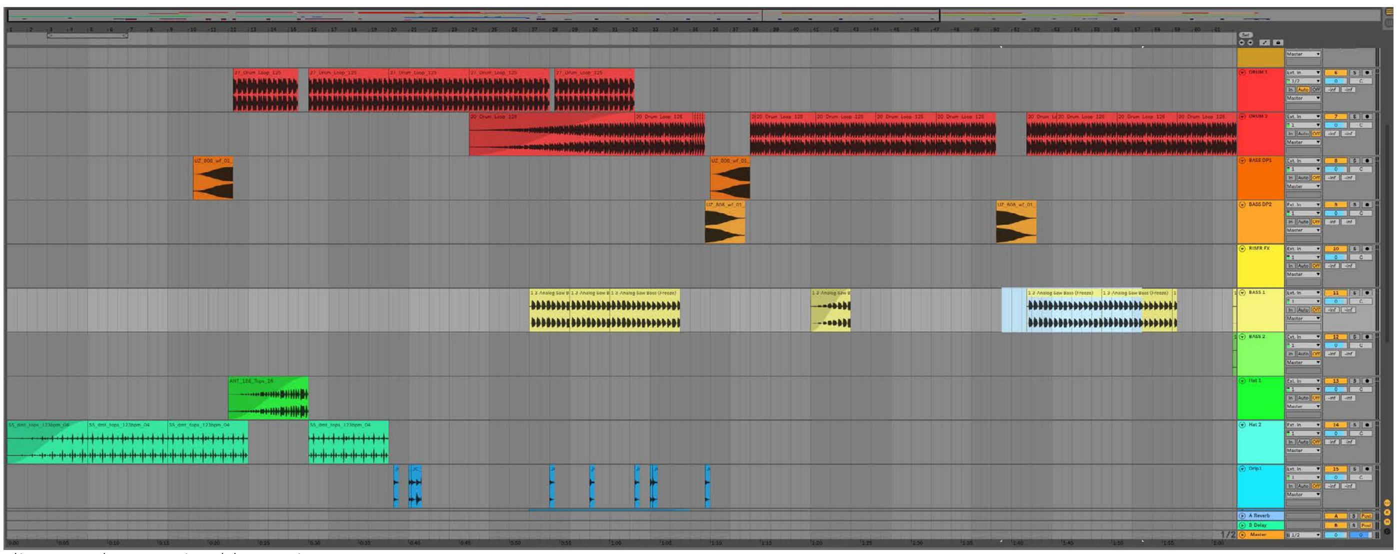


fx



sax

Next variation required another test through completely different set of tracks to prove that the sequence that I've created works for any given composition. As I've mentioned earlier, the song that was used was not an ideal composition because it didn't meet basic requirements of a house track. I've kept a conversation going with Lucas, trying to quickly arrange something that would represent a traditional house track with more clearly defined bassline, drums, drops and vocals. He was able to create a song that I've been using for the rest of the process: "Slippery when wet." Since the previous song was inconsistent in its bass and drum lines, which was vivid in assigned geometries, I was looking for a more consistent composition. A song was created after a discussion where I was explaining what I'm looking for in a song and we've been trying to create a quick beat that was missing a catchy phrase that would become the vocals. I saw a "slippery when wet" sign at that moment which was used as an example: "we need a woman's voice to come in and say "slippery when wet." Lucas made a quick recording of my voice with that phrase and in couple days I had a new song broken down to separate tracks and fitting my expectations.[36] As he continued to work on the song I was able to plug it in my algorithm and generate geometry based on completely new song using the same algorithm.



slippery when wet in Ableton Live

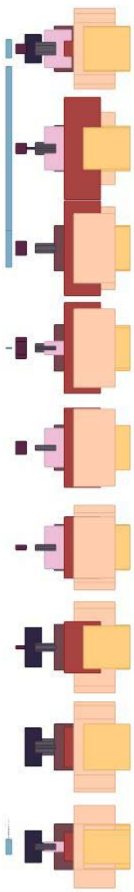
New song contained these tracks:

- drums1;2 – underlying drum tracks
- bass dp 1;2 – bass drops
- bass1;2 – underlying bass tracks
- vocals1 – recorded voice track
- hat1 – hi-hats
- fx – effects

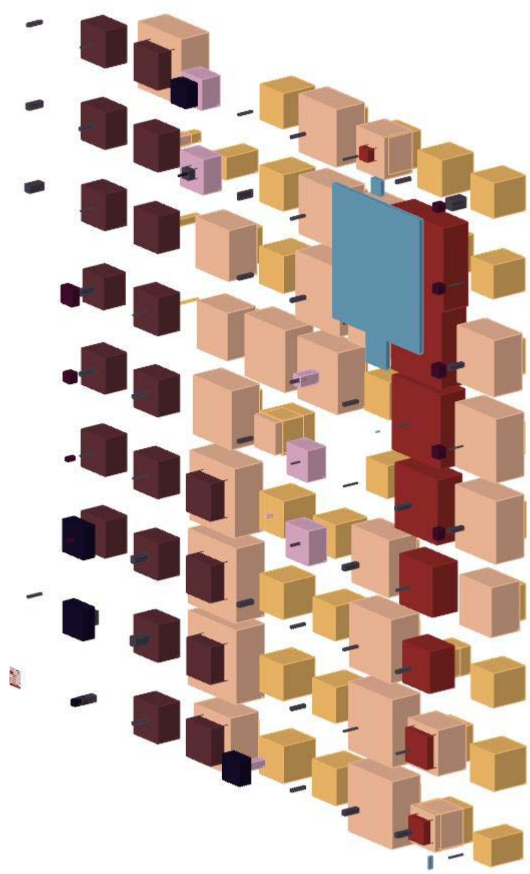
36. Lucas Feazel, personal conversation. September 17, 2018.

Next goal was to eliminate manual adjustments as much as possible in the given time. For example, arrangement of tracks was manual. The loudest and longest tracks were the important tracks and their visual representation had the most complex waveform. Measuring each of those lines would've given me an opportunity to automatically identify the importance of the tracks without looking at them. New components that would measure the length of the tracks were added to the script. Tracks now were sorted by length and arranged automatically level by level. This simplified the process to where any group of tracks can be inserted and the output geometry sorted based on tracks hierarchy.

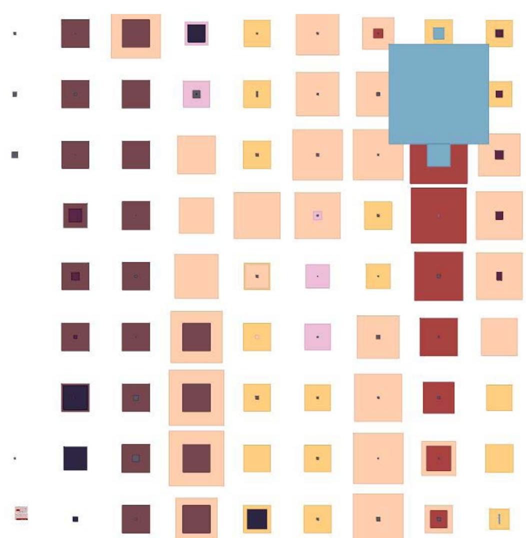
Another variation of the layout was aligning geometry along a curve. Linear array had an outcome directly related to each track, clearly representing each corresponding waveform.



all tracks side view



all tracks axon



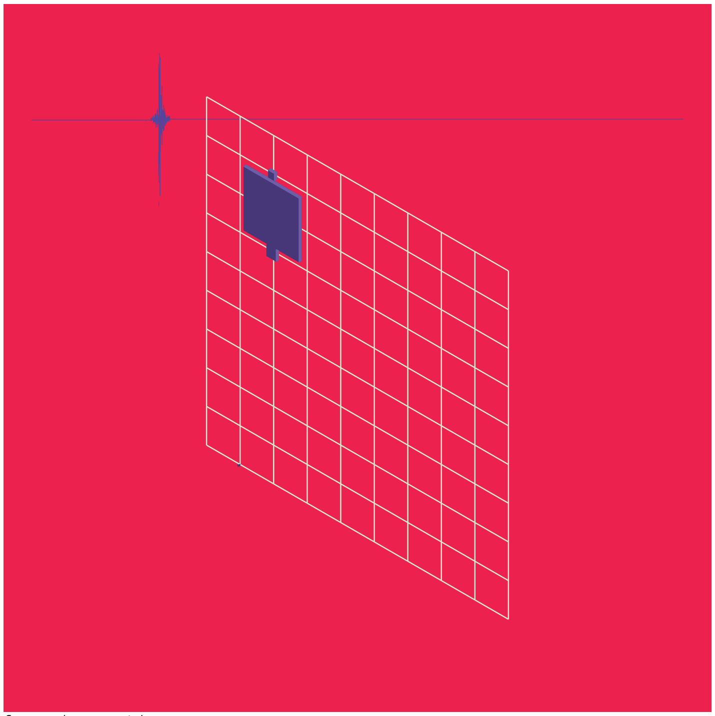
all tracks grid overlay

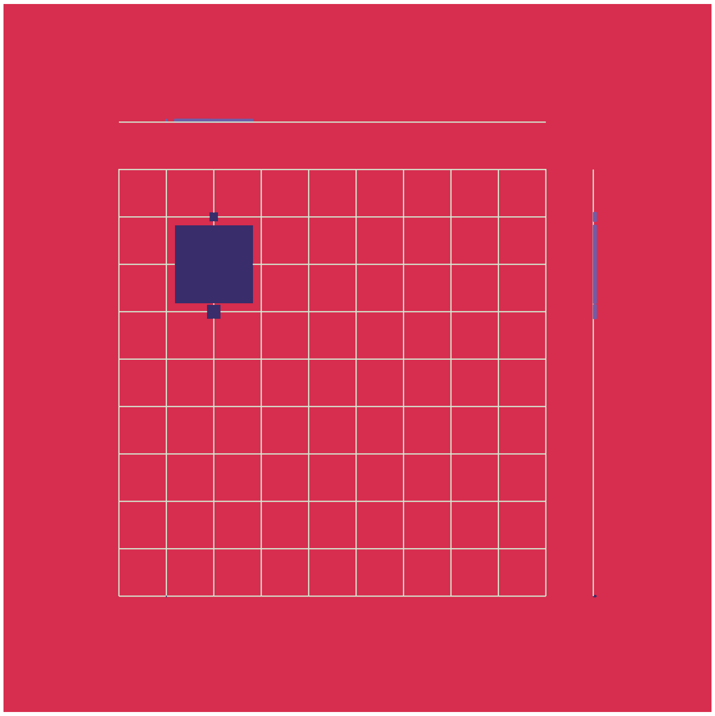
Pseudocode*

```
1.  Generate waveform** from a track
        repeat for each track
2.  Measure waveform's length
        repeat for each waveform
3.  Offset waveforms vertically by length:
        shortest waveform at the bottom
        offset next waveform up
        repeat until done
        compute max value on the waveform
            max value = "a"
            set "a" value as width
            repeat for each waveform
4.  Split waveform into equal segments
        repeat for each waveform
5.  Set a grid based on number of segments
6.  Set segments to correspond with the grid:
        cell 1 = segment 1
7.  Analyze each segment
        compute extremes
        repeat for each segment
        compute distance from (x;y;0) to max point
        on a segment
            compute z factor
            z = distance
            multiply distance by a factor
            compute value "b"
            set value "b" as length
        compute distance from 0 to min point on a
        segment
            multiply distance by factor
            compute value "c"
            set value "c" as height
        repeat for each waveform
8.  Generate form:
        Set a box with width, length and
        height parameters
            width = a
            length = b
            height = c
        Set center of the box to a cell on
        the grid
        Generate a collection of boxes
            Generate a box at cell
                If a>0
                If b>0
                If c<0
        Repeat for each segment at every
        cell
9.  Offset collections of boxes from each other
        Don't offset if: waveform of collec
        tion of boxes located at (x;y;0)v
        Offset by a factor if: waveform
        of collection of boxes located at
        (x;y;z>0)
    repeat for each collection of boxes
```

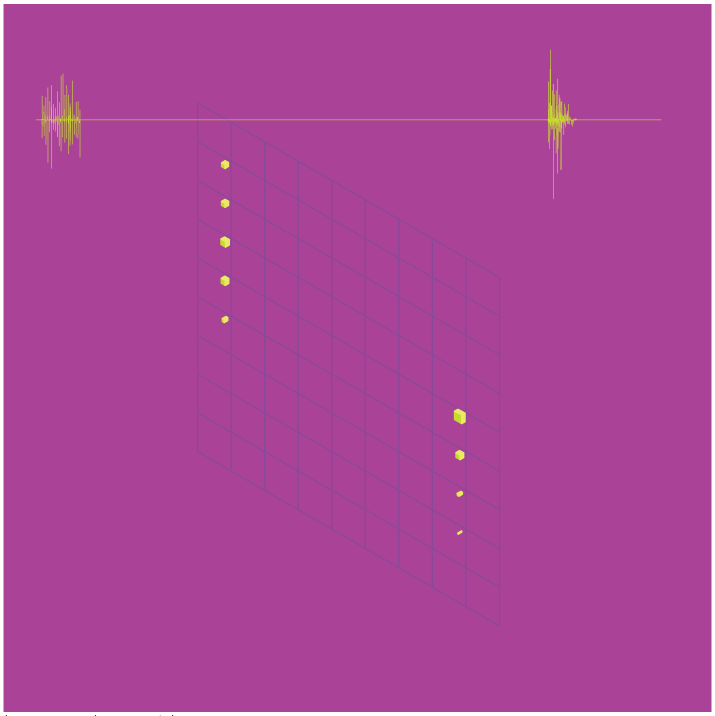
*Pseudocode: a notation resembling a simplified programming language, used in program design
**Waveform: a curve showing the shape of a wave at a given time

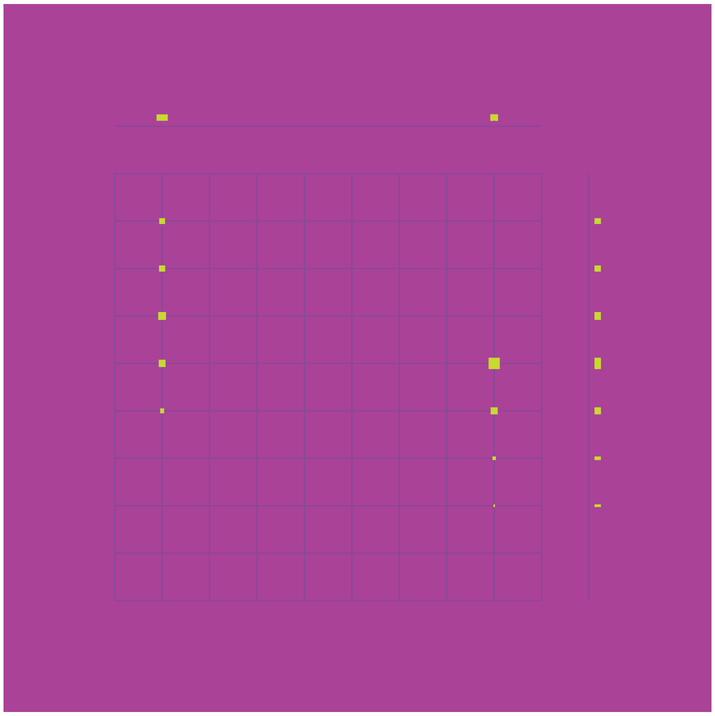
Final output



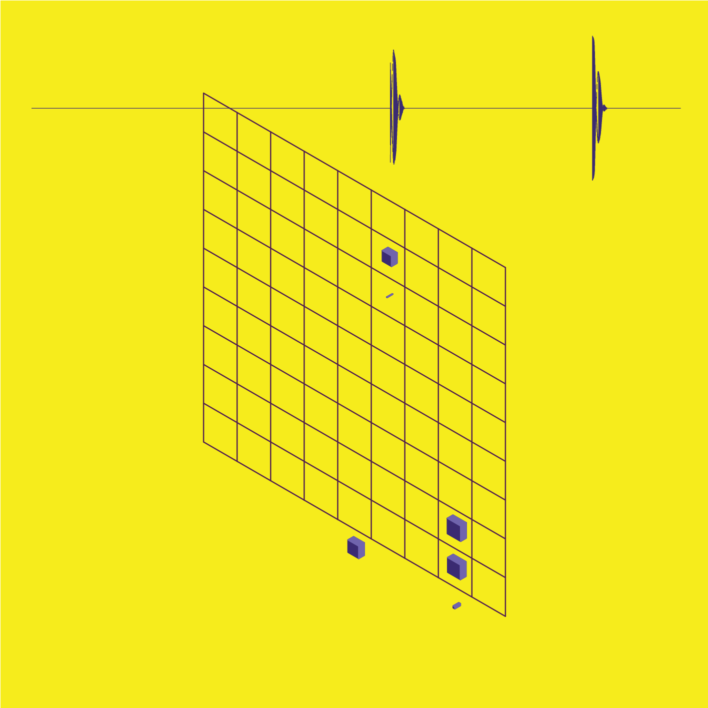


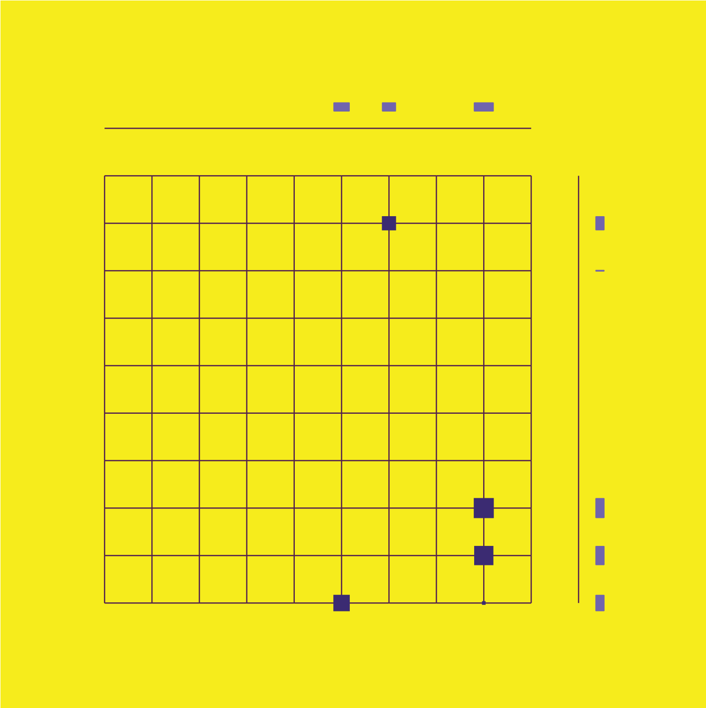
fx track on grid



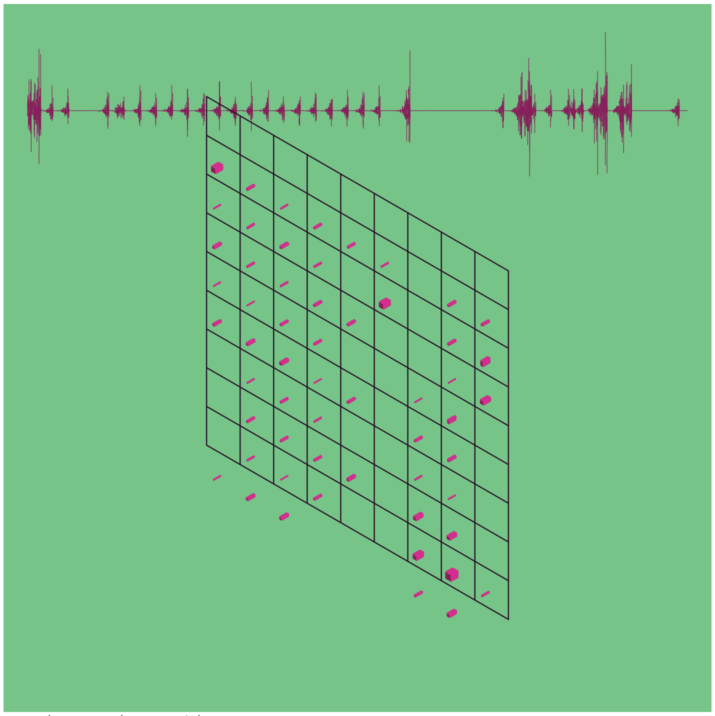


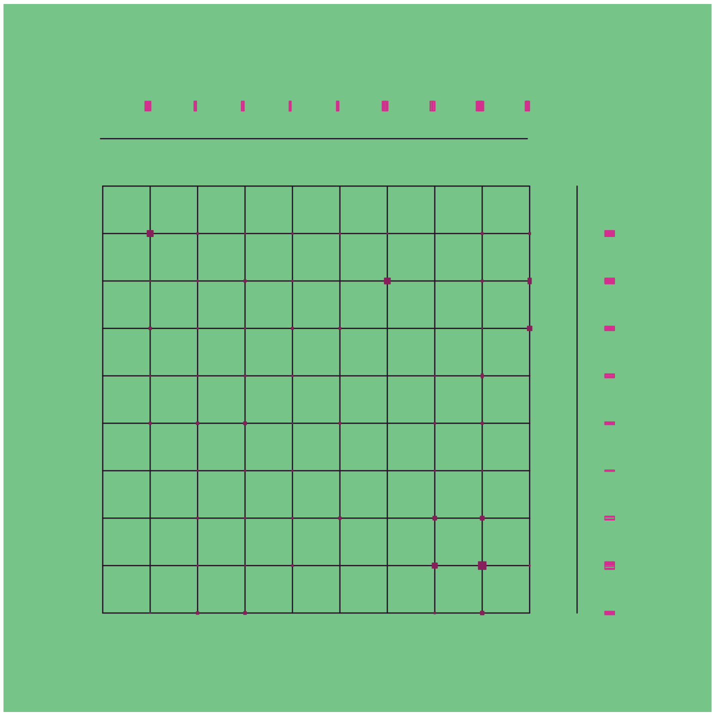
hat 1 track on grid



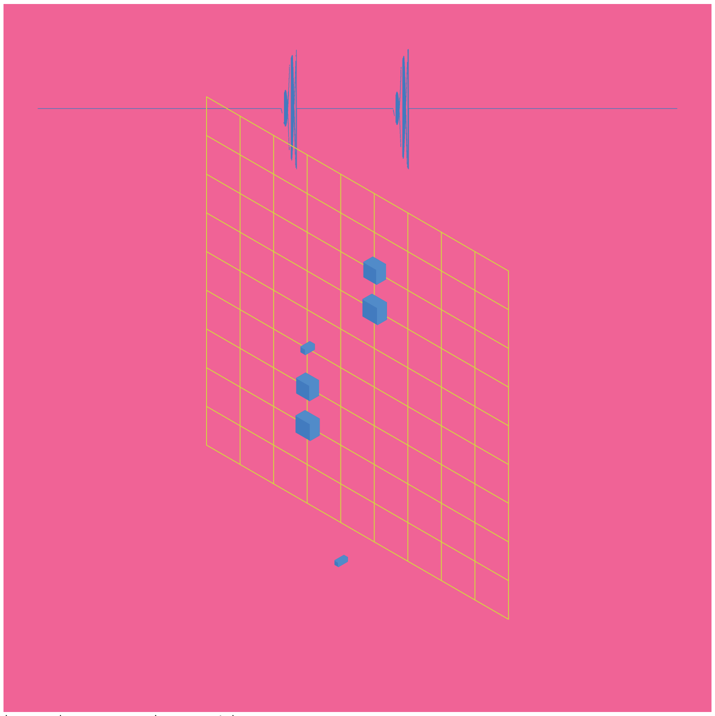


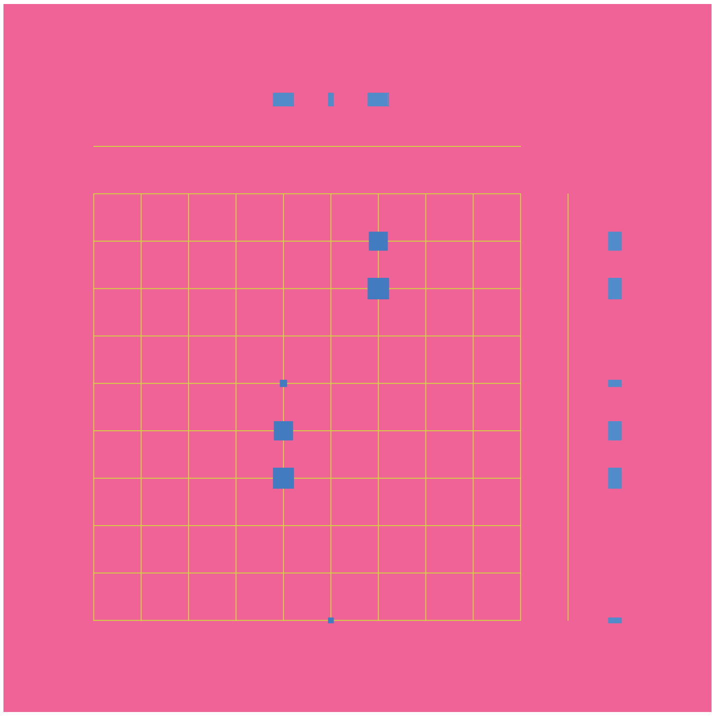
bass drop 1 track on grid



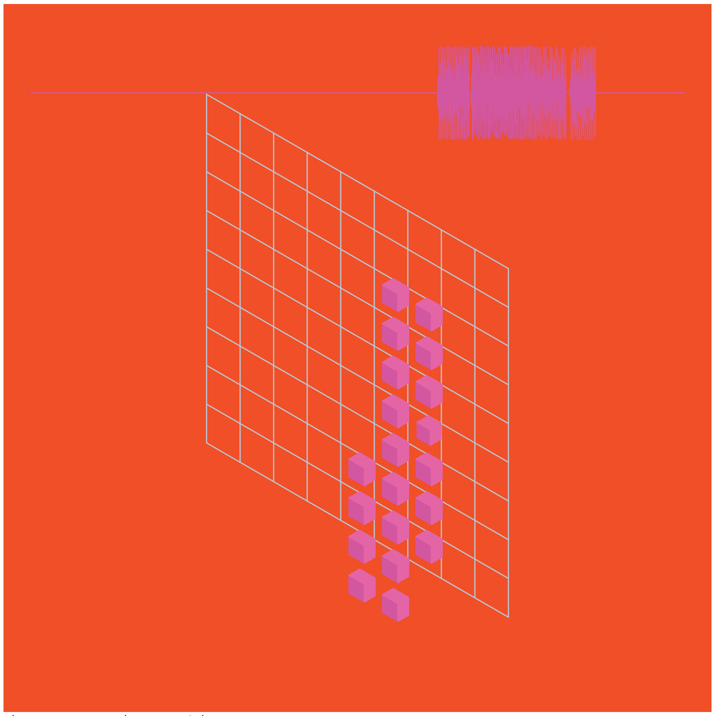


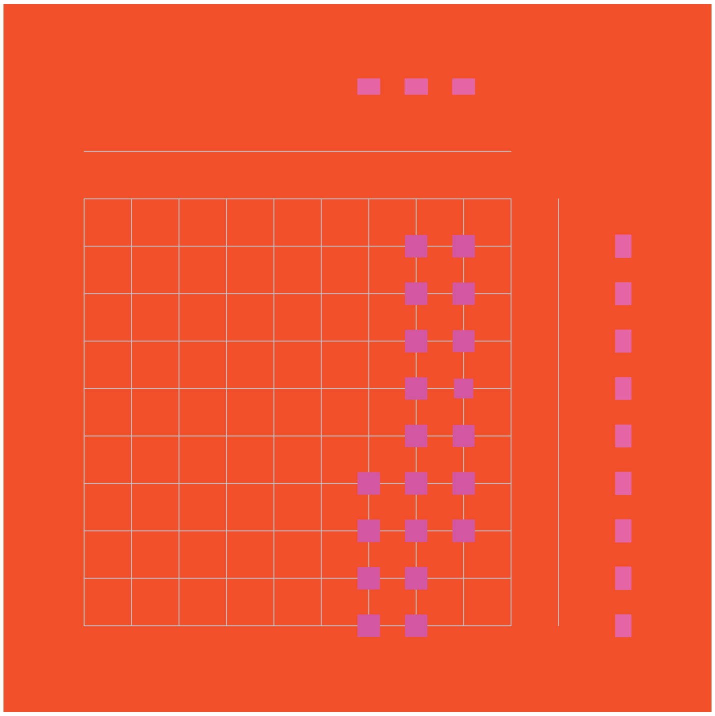
vocals 1 track on grid



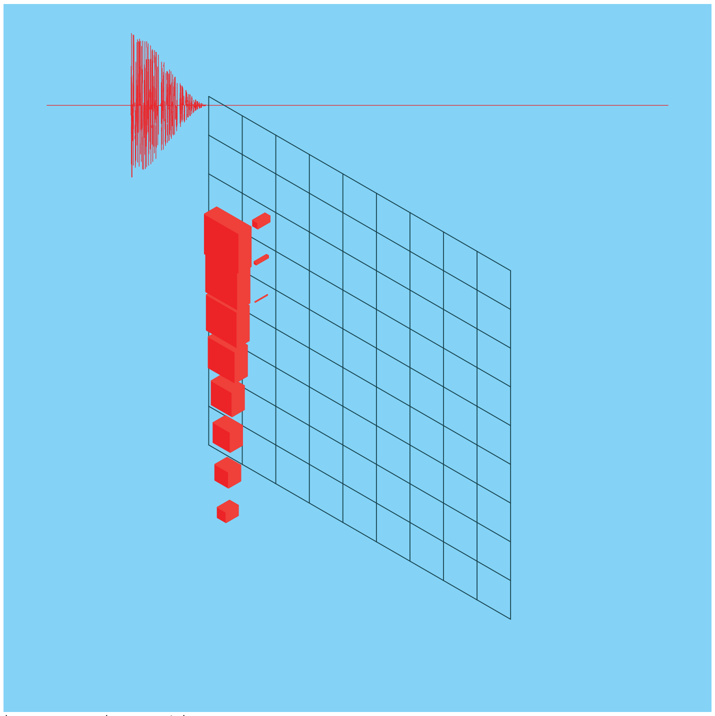
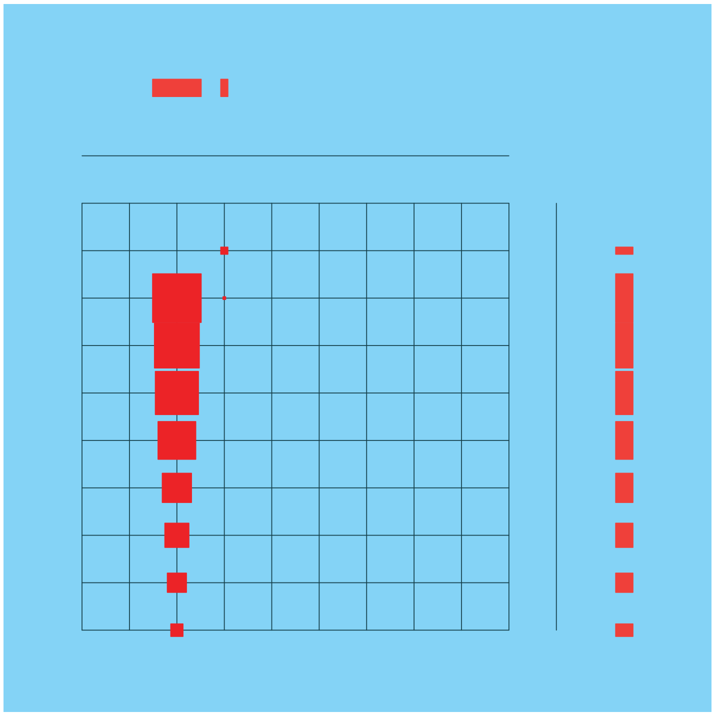


bass drop 2 track on grid

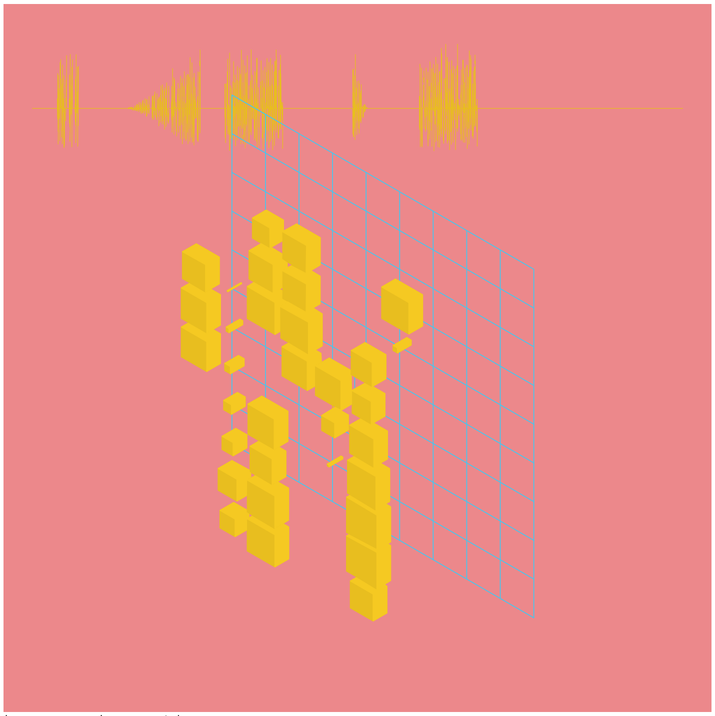
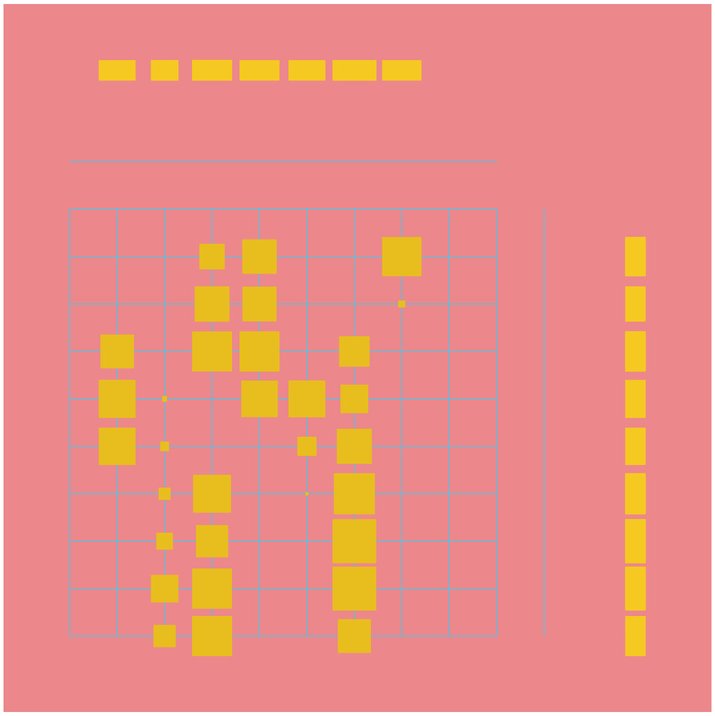




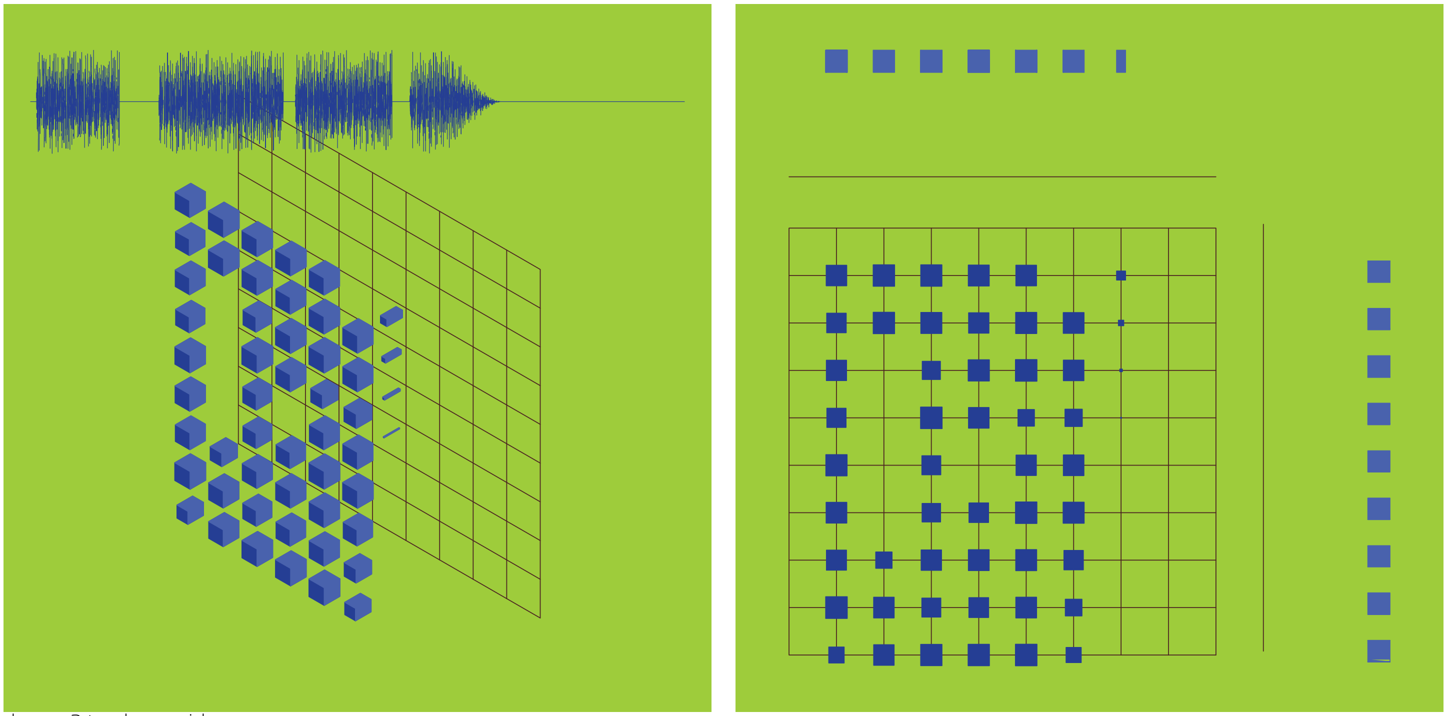
drums 1 track on grid

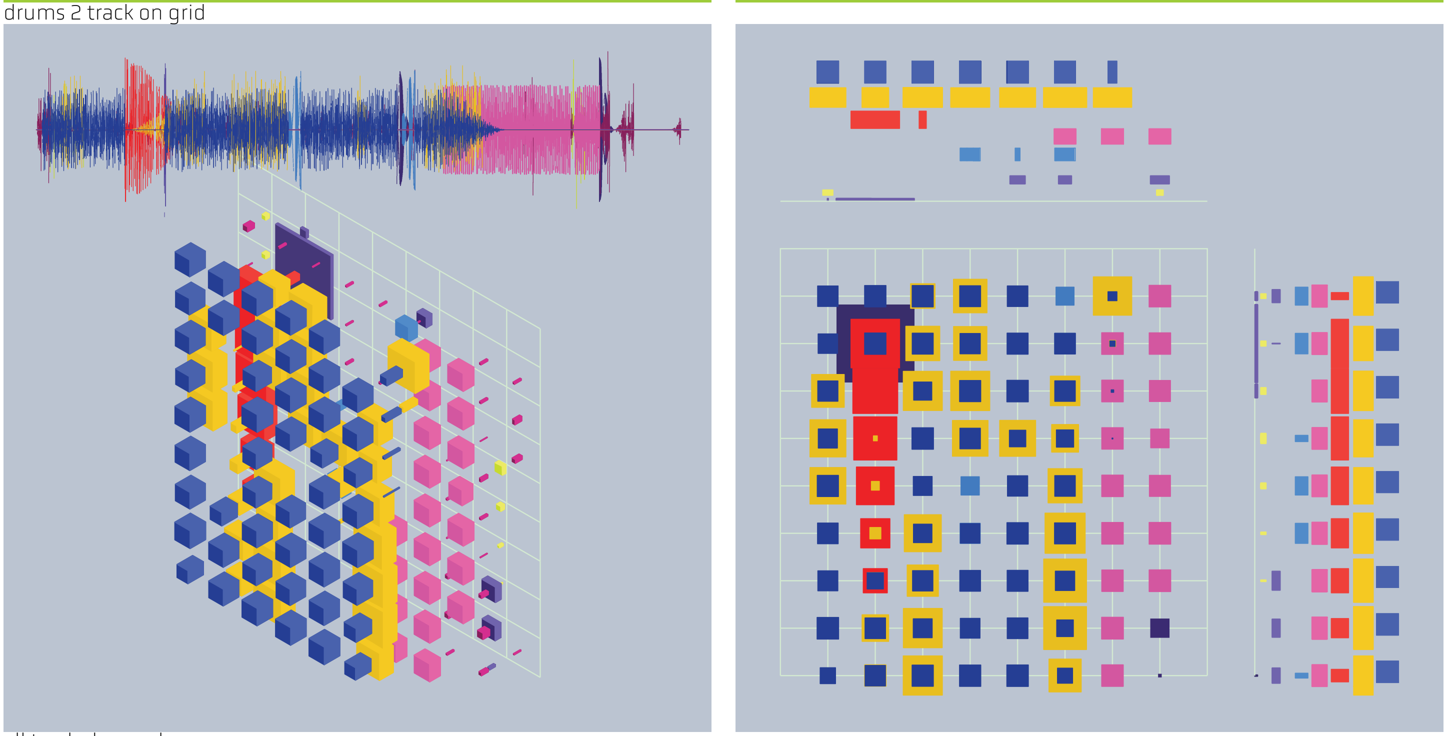
bass 2 track on grid

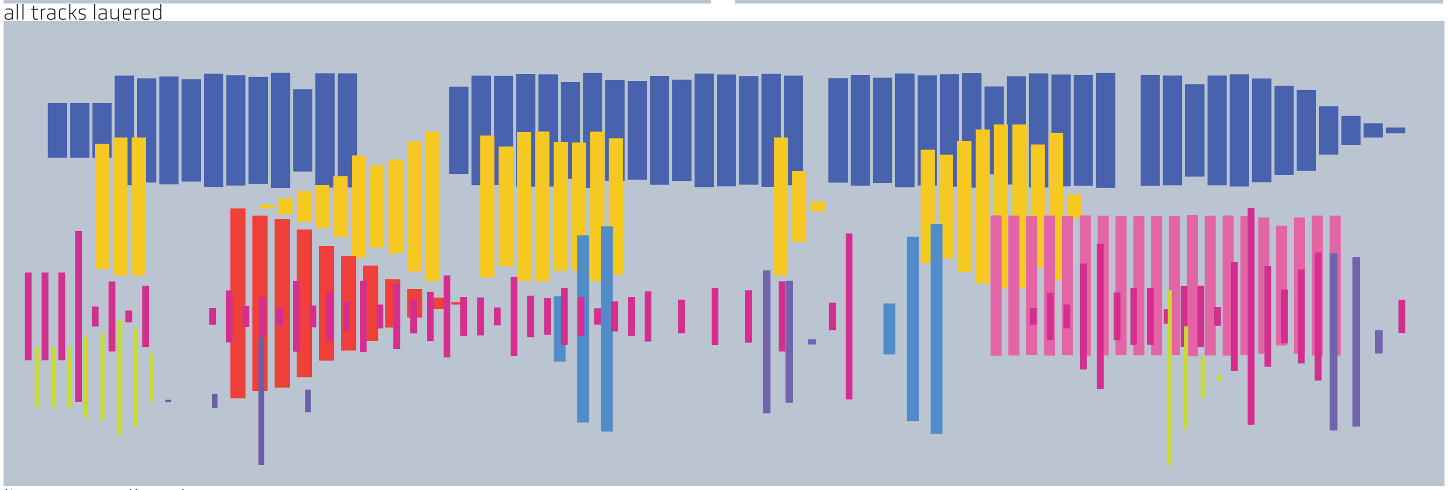
bass1 track on grid



drums 2 track on grid



all tracks layered



linear array all tracks

Conclusion

As a main result of my project I developed an algorithm that can be used with any song of any genre and create geometry based on that song's parameters. Algorithm is functional and can be adjusted based on the user's needs. Final geometry can be aligned to a grid or a linear array; it is flexible and can be based on any assigned forms.

Results prove that music can become 3-dimensional and be used as form-generating tool through given parameters. In addition to geometry, this project inspired two songs that were used and studied here: "Looking Back" and "Slippery When Wet." Songs were tailored to the project based on the theory of house music. Final outcome is an abstract representation of what songs could look like.

The script was created with algorithmic thinking in mind, where data was collected and analyzed through computation, providing a new tool for me to share and use in the future.

The interpretation of the output could lead to creation of space through volumes or voids between them. Any song of any genre can be plugged into the algorithm and produce an output that can be studied in terms of volume and further adapted as an occupiable space.

DISCUSSION

First explorations of computation in architecture involved stylistic approach and formal expression. Today architecture reached a point where design is based on relationships between objects, which is also known as parametric design: once a single component of the system is changed it automatically affects the related components. With the ability to process data and advancement in scripting it became possible to generate a set of variables into a complex form by following a set of rules, or algorithm.[37]

Music has many meanings to many people, it's something that everyone can discuss and relate to on emotional level. Music associates with many things, brings up memories: good, bad, happy, sad etc. Architecture provides a set – where something happens, where people spend their time. Architecture also has emotional appeal; it can be touched and seen. Designers have been trying to apply mathematic rules and ratios, harmonic and rhythmic orders to connect music with architecture in the past, [38] today we have access to tremendous amounts of data, [39] which includes music and its contents. We can manipulate and translate data directly from one source into another. What if we could see music in architecture? How would that architecture look like? How would it feel to be in that space? Space not just inspired by music, but generated directly from music.

My project gives just a tiny glimpse of what parametric tools can be used for and how music could be visualized, but it proves that music can become 3-dimensional. It's an important step in the design process that with further development can affect our architecture.

I chose house music at least for its name – "house" implies architecture, even though the name came from the name of the venue where it was first played.[40] There's more to it than just name, this is parametric music, music of our time created by people with use of algorithms and computation. House music doesn't require one to have anything but a laptop, however it requires a skill of knowing how to combine tracks together.

For centuries we've been creating music from piecing together instruments, rhythms and voices wishing we could improve the quality, find a better sound, know how to play all instruments and save some time. Same with design – from cave drawings to drafting tables to computers – humans were inventing the process, using new tools and improving the quality of work. Music making and design are very similar processes and in this project I've studied music approach and applied it to form making. I've created a sequence that might become useful for both musicians and designers. Visual representation of a song might guide a musician to change the composition and other way around – a designer hears a song that they would want to visualize and it's possible at this stage.

The algorithm could be improved in several areas that would require further studies, such as breaking a song down to tracks without going through additional software and conversions. It's almost identical to flattening layers process: once all layers of a digital drawings are flattened we receive a singular image, same with the song, it becomes singular once it's released from editing software, which makes the layers difficult to extract because they're mixed together.

I'm sure there are ways to recover that data, but it will require further experiments.

37. Sakamoto. From Control to Design.
38. Jenks. "Architecture Becomes Music"
39. Carpo. "Big Data and the End of History"
40. Complex. "A Brief History of House Music"

Generating a space would be a bigger goal for this project, using this algorithm to generate not just form, but architecture with ability to re-adjust parameters and better understand capabilities of parametric design tools. Re-aligning tracks as structure, substructure and other architectural elements would give an opportunity to generate a space that can be constructed and occupied.

One of the ways that this process could generate a space is as if we draw a parallel between each track and align it with architectural elements. The essence of house music is in its bass drums, which represent the foundation of the song, without them the song wouldn't exist. If we make direct translation of drum track to architecture, we receive a foundation that would carry the entire building. Some songs can carry several drum tracks, which could be read as a structure of a building, generating the physical form directly related to form generated by the algorithm.

Besides literal translation of the generated form into direct architectural elements we could look at the composition as a whole in terms of volumes. One of the approaches could be incasing the space in the generated volumes — the program occurs inside of each generated element that have potential to intersect and overlap each other within one or multiple tracks. Another approach would be the space that occurs between the volumes. The cells that remain unoccupied at every track create voids between each volume and can potentially become a program. Generated form could be read as collection of volumes that could carry different program without distinction between the tracks, as long as volumes overlap and create their own unit they can contain the program. Form can also be understood as a space that divides the program by tracks regardless of intersections from one space to another. In this case form is a solid that is sliced through by multiple tracks.

Another way of interpreting the output could be through the opacity of the volumes and studies of intersection between the volumes. In this case volumes could be subtracted from each other at the points of intersection, or create another volume within the intersection. Movement through a space generated by the song could be imagined as a way of collapsing time since each volume is responsible for a segment and each segment carries time aspect in them. If we are to move through the grid diagonally, we would be passing through the moments of the song against its original layout, skipping some of the volumetric elements and experiencing the other sooner than the song intended. Algorithm has potential to introduce another way of generating music by giving feedback to creators of songs. Based on the output we can observe the areas that could benefit from additional volumes or need volumes subtracted giving an opportunity to provide not only great sound but also more dynamic composition through working back and forth with artists.

On smaller scale the output could be used to generate smaller objects. Jewelry is one of the examples — each piece of the song can be 3-D printed into a string of volumes that could become a bracelet or a necklace, particularly in linear array. Since the algorithm can work with any song, generated piece could be very personal and carry any song in its foundation.

BIBLIOGRAPHY

Carpo, Mario. “Big Data and the End of History.” International Journal for Digital Art History. Accessed November 29, 2018. http://nbn-resolving.de/urn:nbn:de:bsz:16-dah-499134.

“A Brief History of House Music.” Complex.com. Accessed December 09, 2017. http://www.complex.com/music/house-music-history.

Felton, David. “Deconstructed: Caribou – Can’t Do Without You.” Attack Magazine. 2015. https://www.attackmagazine.com/technique/deconstructed/caribou-cant-do-without-you/

Feazel, Lucas. “Looking Back”. 2018 MP3

Feazel, Lucas. “Remedy” Mike Mago. Remix. 2018 MP3

Feazel, Lucas. “Slippery When Wet”. 2018 MP3

Feazel, Lucas. personal conversation. September 17, 2018.

Goulthorpe, Mark, Amanda Reeser, Ashley Schafer, and Alayna Fraser. “Precise Indeterminacy Three Projects by DECOI and an Interview with Mark Goulthorpe.” PRAXIS: Journal of Writing Building, no. 6 (2004): 28-45. http://www.jstor.org/stable/24329184

“Grasshopper.” Grasshopper. Accessed November 29, 2018. https://www.grasshopper3d.com/.

Hydlide. “Basic Elements: House Music.” Reason Experts. 10.2016. https://www.reasonexperts.com/basic-elements-house-music.html

Jencks, Charles. “Architecture Becomes Music.” Architectural Review. 6 May, 2013. Accessed December 09, 2017. https://www.architectural-review.com/rethink/viewpoints/architecture-becomes-music/8647050.article.

Locke, John. “Sound.” Gracefulspoon.com (web log), 2012. Accessed December 10, 2017.

Morris, Colin. “6 Weird Pop Songs Visualized.” March 14, 2017. Accessed December 10, 2017. http://colinmorris.github.io/blog/weird-pop-songs.

Peters, Brady. “Computation Works: The Building of Algorithmic Thought.” Architectural Design 83, no. 2 (2013): 8-15. doi:10.1002/ad.1545.

Reilly, Chris. “Data Tree Components.” Lynda.com - from LinkedIn. September 26, 2014. Accessed November 29, 2018. https://www.lynda.com/Grasshopper-tutorials/Data-Tree-components/174491/194128-4.html?autoplay=true.

Reilly, Chris. “Grasshopper Essential Training.” Lynda.com - from LinkedIn. October 26, 2018. Accessed November 29, 2018. https://www.lynda.com/Rhino-tutorials/Grasshopper-Essential-Training/599608-2.html

Rutten, David. “The Why and How of Data Trees.” Grasshopper. Accessed November 28, 2018. https://www.grasshopper3d.com/forum/topics/the-why-and-how-of-data-trees

Sakamoto, Tomoko. From Control to Design: Parametric/Algorithmic Architecture. Barcelona: Actar, 2009.

Smuts, Carson, Breanna Carlson. “Mosquito.” Studio Smuts. Accessed Nov. 29 2018http://www.studiosmuts.com/ceed3/mosquito/

Tedeschi, Arturo. Algorithms Aided Design. Potenza, Italy 2017.

Wood, Sidney. “Understanding Waveforms.” SWPhonetics. September 12, 2018. Accessed September 16, 2018. https://swphonetics.com/praat/tutorials/understanding-waveforms/