Towards Improving Collaboration Between Visualists and Musicians at Algoraves

Zoyander Street

Lancaster University
z.street@live.lancs.ac.uk

Alejandro Albornoz

University of Sheffield
akousma.viva@gmail.com

Renik Bell

Independent
renick@gmail.com

Guy John

Independent
johnguy@rumblesan.com

Olivia Jack

Independent
rhizomaticode@gmail.com

Shelly Knotts

Durham University
knotts.shelly@gmail.com

Alex McLean

Deutsches Museum
alex@slab.org

Neil C Smith

Independent
neil@neilcsmith.net

Atsushi Tadokoro

Keio University
tadokoro@gmail.com

J Simon van der Walt

Royal Conservatoire of Scotland
js.vanderwalt@rcs.ac.uk

Gabriel Rea Velasco

Independent
gareve25@gmail.com

ABSTRACT

This paper arises from a group discussion took place at the 2018 Livecoding Moot in Sheffield about the role of visualists in algoraves, with the particular aim of improving collaboration and giving visualists a more equal status both conceptually and practically. The current situation is one of imbalance, in which visualists are often relegated to a secondary role. Some suggestions for improving the situation are examined, including challenging certain social practices, and employing technical approaches aimed at fostering a more balanced collaboration.

1. INTRODUCTION

Algorave events often bring together two distinct groups, working in different media: visual artists and musicians. These two groups don't always share a language for describing their practice and often struggle to coordinate their two mediums. Closer collaboration and more harmony between music and visuals could go some way to resolving this, as well as creating performances with clear continuity between both music and visuals.

The present paper reflects on a semi-structured group discussion attended by livecoding musicians and visualists at Livecode Festival in Sheffield UK on 3rd September 2018, during which participants collaborated on looking for ways to establish 'best practice' for algorave organisers and participants, and how norms in audio, visual and audio-visual performance can be challenged and reconfigured. Notes were taken using mind-mapping methods on a long roll of paper, to visualise and spatialise the discussion as it went on and to form a starting point for this paper. After creating notes based on this record of the discussion, some participants collaborated on co-authoring the current paper.

Towards this aim the following paper introduces background to the discussion, highlights problems in how algoraves are organised, and puts forward ways in which these different groups can work together in the future. Specifically, we point toward: social practices that might improve collaboration; technical methods of improving synchronicity between visuals and audio; and proposed or recently-developed practices that return to the shared foundation of live coded visuals and audio in the computer platforms themselves, and might offer means of collaboration that offer simplicity, coherence, and an authenticity to the material.

2. BACKGROUND

The tongue-in-cheek 'motto' of the livecoding scene 'show us your screens' implies that there is always a projected visual element to a musical live coded performance. It has become common practice for algoraves to combine both music and visuals, with a two-person team of one livecoding musician and one livecoding visualist being a ubiquitous model. Despite a long and well-documented history of audiovisual artworks (see Ribas 2014, Correia and Tanaka, 2014), relatively little discussion of this particular collaborative practice is to be found in the recent livecoding literature. Shawn Lawson and Ryan Ross Smith describe a performance 'Kessel Run' (Lawson and Smith, 2014) in which they co-improvise GLSL shaders and fragments of pre-composed music in a single browser-based environment, while Dagobert Sondervan and Kasper Jordaens (2015) have presented their collaborative work as H.AL.I.C. (Heuristic ALgorithmic Interactive Computing), generating real-time images and music in a complex environment built using Clojure, Overtone, Qil, Processing and SuperCollider.

Algorave inherited an imbalance between musicians and visualists as a hangover from 90s club culture, where VJs were added as a complement to DJs. Given the focus on performativity in algoraves, we wanted to look at ways to redress this balance, in a similar way to how initiatives such as LPM – Live Performers Meeting (LPM) (2004) have sought to enhance and promote veejaying culture. A club event with dancing might be seen as sufficient with just music, and culturally there are not so many events where people just dance to silent visuals. However, this discussion was about algoraves and grew from the view that both are an essential part of the event.

Compounding this problem, in some algorave cultures, there is a shortage of visualists, so a small number of visualists are trying to accompany a large number of musicians. To take a couple of examples of online communities: of the dozens of artists listed as 'friends' on the algorave.com website, and judging on the information available via the linked pages, it would appear that only three are mainly or exclusively visualists. The rest are all either musicians, or musicians with an expanded practice that includes visual approaches. While, on the talk.lurk.org server, there are currently 830 users in #visualists out of a total membership of #general of 2033: 69 users in #hydra versus 1223 users in #tidal.

The algorave at DiNa in Sheffield can serve as a good example of a typical algorave. Of the thirty-two artists who participated only four were solely visualists. In this case, Bruce Lane had already agreed on a performance with a musician and only wished to perform a single set. This meant that the remaining three visualists, hellocatfood (Antonio Roberts), Wispy (Will Humphreys), and rumblesan (paper author Guy John) were responsible for the visuals for the rest of the event, across the two rooms. Organisation of this almost entirely consisted of ad hoc discussions just before the end of the currently playing musicians sets, and then tracking down the next performer to ask if they would or would not like visuals. It was also noted that the visualists were also not promoted in the same way as musicians on advertising for the event, nor listed on the set lists available to the audience. Both are problems that the paper authors felt needed to be addressed.

3. SOCIAL PRACTICES

3.1. Addressing disciplinary imbalance

It would seem important for organisers of algorave events to take specific steps to better accommodate visualists. A simple practical point is to plan for good quality projectors and planned projection surfaces to be made available for the visualists as well as the livecoding musicians. The visual artists should be advertised and promoted alongside the musicians: this would in itself give stimulus to more visualists to take part in algoraves. There should where possible be transparency in the planning, so that performers in the respective disciplines have the opportunity to organise their work together in advance.

3.2. Improving communication

It was noted in the discussion that there might be a lack of shared language between visualists and musicians, making it difficult for musicians to ask for what they need from visualists. Should communication between these two disciplines defer to well-defined musical terms like rhythm, bar length, and other musical structures? Or, to equalise the relationship between visuals and music, should it instead be based on emotions, conceptual ideas, or other abstract terms?

Collaboration between visualists and musicians in an algorave is often last-minute. Making contact prior to the event could enable shared planning of a set – when will there be a change in the music, how do we plan for a drop? As a simple example, in preparation for the Sheffield algorave, coder-musician tedthetrumpet (aka paper author J Simon van der Walt) contacted visualist Bruce Lane on the Rocket.Chat server talk.lurk.org to discuss the possibility of working together. In subsequent online discussions the duet were able to build a shared trust in their individual practices by reference to, respectively, previous musical performances documented on YouTube, and examples of possible visual styles on Shadertoy.

Musicians noted that they often found it hard to look at visuals when they are focused on creating code. However, there may be ways of mitigating this issue by having a monitor or similar of the visuals within the musician's eyeline – in Sheffield a number of musicians mentioned they were more aware of the audience than the visuals, at least somewhat because they are directly in front.

Visualists try to anticipate what the musician is going to do next, but this is not always easy – to some degree, it would be easier for visualists if musicians kept to a rigid structure and played by the rules of 8-bar phrases! In practice there is often a such wide breadth of styles and influences being performed that this rarely happens. For instance, a livecoding musician with a background in Javanese gamelan music might use articulatory structures such as a change of irâmâ (note density) or laras (scale). However, a visualist unfamiliar with these gestures might not understand what was about to happen in the music, and, more importantly perhaps, when. Sharing this kind of knowledge is one of the pleasures of collaboration of course, and communicating about these kind of topics in advance of events, rather than relying on ad-hoc pairings of musicians and visualists, would help to prepare artists, so that they can collaborate with a common vision and create more coherent works. Direct agreement between musicians and visualists is one way to make this common ground, or organisers can also suggest it.

In addition to the social solutions described above, a number of technical solutions for visualist-musician collaboration were explored in the discussion at Livecode festival. These are organised below into two categories: methods based on observations of the other artist's final output, and methods based on synchronised systems.

4. TECHNICAL METHODS

4.1. Observing final output

4.1.1. Visualists taking in audio data

Many visualists' platforms can take in audio data and analyse it using FFT to make visuals respond to music. This has been a classic technique in visualizations used by music players such as Winamp.

Some platforms for making visuals that allow this include:

- Jitter (Max/MSP)
- GEM (Pd)
- Processing
- Hydra
- Cyril
- LiveCodeLab
- Fragment
- p5dirt
- openFrameworks

From the point of view of a visual performer, FFT input data can make it much easier to make the visuals seem connected to the audio, without the visualist having to pay constant attention to changing details of the visualisation so that they are congruent with musical changes. By using data about the lower frequencies created by the kick drum for example, it is possible to create changes in the visuals that appears to be "in time" with the music, without having to try and manually beatmatch or sync systems.

In the context of algorave this can be good and bad however. Given the experimental nature of a lot of the music, with potentially changing tempos and time signatures, beat matching may be extremely difficult, and so FFT may be the only reasonable way to get this syncing. Conversely, in musical performances that have much denser patterns and structures, this kind of input may end up appearing quite noisy and random. As a result, more thought might be required regarding the frequency bands used or ignored.

One of the larger downsides with the direct audio input is that it can be very difficult, if not impossible, to automatically discern the higher-level patterns that may occur within a musical performance. A visualist following a musician may be able to detect ahead of time that there's a drop coming up and plan a visual change for it, but doing this automatically based of an audio feed feels like a very hard problem. A beneficial result of this is that the visualist retains independent creative responsibility for interpreting the structure and meaning of the music, with the audio data enhancing the visual performance but not determining it.

4.1.2.Musicians taking in visual data

A technique that is possibly less used at present is for musicians to use data generated by visualists in the generation of music. There are fewer readily available tools, and there is often a high technical or CPU overhead to do real time graphics analysis. Given the lack of standard practice and tools, the discussion here suggests some possible approaches and examples.

One suggestion was that a grid could be used to analyse the visualists' output and interpret hue, texture, intensity etc. for different areas and use that data to feed variables in music code, acting somewhat similar to hardware controllers like the Monome. Differences between one frame and the next at grid locations could be used as triggers, or x/y motion tracking of elements within the visuals could act as variables available within the musician's code. At ICLC 2018, the performance by Yosuke Sakai and Hiroto Takeuchi, "Improvisation in Painting And Composition" did this: a sumi-e painting was used as a spectrogram representing time, frequency and amplitude, that was then further modified by the musician.

Fragment is an example of a platform that generates sound based on the interpretation of visual data. As a "collaborative, cross-platform, audiovisual live coding environment", it takes a "pixel-based real-time image-synth approach to sound synthesis". That is to say, live GLSL code is used to produce pixel data on the graphics card, and the sound synthesis uses parts of the visual output ("fragments" made of one-pixel wide captured areas) to influence the sound.

During the discussion at Livecode festival, it was proposed that generating sound based on visual data works better when the tonal qualities are being influenced by visuals, rather than the musical structure itself, which audiences typically expect to more or less conform to the familiar patterns of house, techno, etc. (4/4 time, 8-bar structures etc.).

How else could we allow visualists to change the state of audio? One possibility would be for the livecoding musician to receive a copy of the actual program text being coded by the visualist. There might then be ways for the musician to re-parse this text into the context of the musical language. As a thought experiment: a visualist codes the following text in Hydra:

```
osc(20, 0.01, 1.1).kaleid(5).color(2.83,0.91,0.39).rotate(0, 0.1).scale(1.01).out(o0)
```

A musician working in SuperCollider receives this text and, for instance, extracts any numbers within parentheses to be incorporated into the musical texture:

```
Pbindef(\foo, \note, Pseq([20, 0.01, 1.1, 5, 2.83, 0.91, 0.39, 0, 0.1, 1.01],inf), \dur, 1/4).play
```

Or, a possible reinterpretation in FoxDot, using just the characters from the Hydra code:

```
d1 >> play("osckaleidcolorrotatescaleout")
```

4.2. Syncing systems

4.2.1.Exchanging messages in code

These analogue-ish methods based on extracting data through analysis of the other livecode artist's output can have attractive aesthetic qualities. However, they add an extra layer of engineering and data generation to a collaborative system that is already rooted in code and data – at the discussion at Livecode festival, some participants expressed an interest in instead using the code data that already exists in common between visual and musical livecode methods, for a more elegant and direct form of collaborative performance.

As an example, OSC is a well-established communication protocol which can been used to link visualists and musicians, allowing a visualist to make changes to the audio, and a musician to make changes to the visuals. In the context of live coding this allows a flexible method to send state changes or other messages to a collaborator's performance system. Its use is not particularly widespread among collaborations between visualists and musicians at algoraves at the moment, so we recommend it here as a possible solution.

For example, Renick Bell and Joana Chicau have implemented collaboration using a Haskell OSC library and OSC.js in the browser, allowing each system to affect the other by sharing messages which trigger state-changing functions (Bell and Chicau, 2018). Alex McLean (yaxu) and Rodrigo Velazquez (yecto) have implemented a system that takes OSC messages from Tidal to influence visuals rendered through Processing, called “s2hs2”. Bell and Atsushi Tadokoro have also used OSC for audio-visual collaboration. Tadokoro prepared in advance 100 pattern shader (GLSL) animation sequences and five types of post-effect shaders. OSC messages that change the shader or effects were decided upon before the performances, and those visual changes were synced to the musicians’ changes. This system also made it possible for the visualist to intervene into the musician’s system. There could also be methods by which the visualist could change the tempo and patterns used in the musical performance with OSC messages that have been decided upon beforehand.

Of course, suggesting OSC *in itself* is neither novel, nor perhaps the full answer to our question. The flexibility of OSC over more limited and rigidly defined protocols like MIDI is both a benefit and a problem (with various suggestions for standardized namespaces throughout its history). We suggest that one solution might be a common grammar or specification for bi-directional OSC communication between live-coding musicians and visualists, building on top of existing practice. This would allow suitable mappings between software to be pre-prepared – collaborators would just have to agree what they were responding to, for example:

```
/livecode/bpm 160
/livecode/tick *
/livecode/pattern/1 “xx-o[---]”
/livecode/xy/1 0.5 0.2
/livecode/color/2 0.8 0.2 0.7
/livecode/mood “purple”
/livecode/say “Thank you for choosing algorithms”
```

4.2.2.Sharing a foundation

During the discussion at Livecode Festival, Akihiro Kubota pointed to the value of returning to the root of livecoding of both visuals and music in code and data. Participants could share a common foundation to a performance, rather than adding extra data through shared messages or extracting extra data through analysis. That is, the visualist and musician could both work on a shared foundation of data that is then expressed both visually and musically. As an example, paper author Shelly Knotts (2017) invited algorave artists to make dance music and visuals out of chemical data deriving from Molecular Soundscapes, a collaborative project exploring the structure of proteins through sound.

5. THE AESTHETICS OF INCOMPLETE SYNCHRONICITY

Much of this paper has explored methods for increasing the synchronicity between audio and visual aspects of a collaborative performance. However, as has occasionally been hinted above, there was very little appetite during the discussion at Livecode festival for a situation where the visuals completely match the audio or vice-versa.

Unsynchronisation might also be an important part of the relationship between visuals and sound. According to Michel Chion (1999, pp. 289-291), the rhythm and ultimately the structure of a film (and subsequently any audiovisual work) is based on “phrasing”, which is structured by “synchronisation points”; these points are moments in which visual and audible events coalesce. This concept assumes that synchronicity between the visual and sonic elements of an audiovisual piece is not constant, but coalesces at these synchronisation points. The periodicity and location of these points give rhythm, phrasing and texture to the audiovisual flow. This feature of what Chion calls “audiovisual coupling” is interesting to bear in mind when looking for creative strategies in any audiovisual piece, including real-time improvisation and live-coding sessions.

Human beings tend to search for and create links between elements, merging all kind of stimuli in a compound perceptual act. If audio and visuals match perfectly, audience attention might subside, as it feels like the viewer can completely understand what is happening once they recognise the pattern. However, if the performance goes in and out of sync, then this could create a sense of tension and release, which could be key to setting pace and maintaining attention. Collaboration between visualists and musicians can create points of synchronisation that re-anchor the performance at key moments, followed by a period of drifting apart.

6. CONCLUSIONS

We make the case in this paper that collaboration between visualists and musicians can improve the aesthetic coherence of an algorave performance, and explore the scope for achieving this through social practices such as addressing disciplinary imbalances and improving communication, and technical solutions that could involve the observation of final output (visuals designed to automatically respond to audio, or music designed to automatically respond to visuals), or having systems synchronise with one another by sharing selected messages or sharing a common data foundation. Though some of this may seem obvious, it has not been standard practice in collaborative performances at algoraves and as a result these suggestions have been listed.

Finally, we reflected on the tension between the desire for synchronicity between visuals and audio, and the recognition that a certain degree of chaos is part of the aesthetic of live-coded music and algoraves. Michel Chion's notion of "audiovisual coupling" is brought to bear in a brief discussion of the temporalities of sound and image working in tandem, and the importance of incomplete synchronicity as part of the artists' expressive palette.

6.1. Acknowledgments

Special thanks are extended to all of the participants in the discussion at Livecode Festival, who in addition to the authors of this paper included: Ryan Kirkbride, Thorsten Sideboard, Chiho Oka, Evan Raskob, Claude Heiland-Allen, and Gabriel Rea Velasco. This paper was made possible by the Livecode Festival event, which was hosted at Access Space in Sheffield with the sponsorship of the Daiwa Anglo-Japanese Foundation, the Sasakawa Foundation, Arts Council England, and the European Research Council.

REFERENCES

Bell, R., Chicau J. 2018. "A Trans-Disciplinary Tool for Collaborative, Choreographed, and Embodied Audio-Visual Live Coding", in: *Proceedings of the International Conference on Live Interfaces 2018*. Porto, Portugal. http://www.liveinterfaces.org/2018/#papers

Bronowska, A., Knotts, S., 2017 "Molecular Soundscapes" https://shellyknotts.wordpress.com/projects/molecular-soundscapes/

Chion, M. 1999. *El Sonido: Música, cine, literatura…*. 1st ed. Barcelona: Paidós Ibérica, Ediciones.

Collins, N., McLean, A. 2014. "Algorave: A Survey of the History, Aesthetics and Technology of Live Performance of Algorithmic Electronic Dance Music", in: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths, University of London, London, United Kingdom, pp. 355–358.

Correia, N.N., Tanaka, A. 2014 "User-Centered Design of a Tool for Interactive Computer-Generated Audiovisuals", in: *Proceedings. Presented at the Inter-Face: International Conference on Live Interfaces*, Sussex, pp. 86–99.

Lawson, S., Smith, R. 2014 "Kessel Run", in: *Proceedings. Presented at the Inter-Face: International Conference on Live Interfaces*, Sussex, pp. 350–351.

Livecode Festival #2 [WWW Document] (2018) http://livecode.toplap.org/2018/ (accessed 9.8.18).

Magnusson, T. 2014. "Herding Cats: Observing Live Coding in the Wild". *Computer Music Journal* 38, 8–16. https://doi.org/10.1162/COMJ_a_00216

Ribas, L. 2014. "On Performativity: Audiovisual Systems as Aesthetic Artifacts", in*: Proceedings. Presented at the Inter-Face: International Conference on Live Interfaces*, Sussex, pp. 72–85.

Sondervan, D., Jordaens, K. 2015 "Heuristic ALgorithmic Interactive Controllers - H.AL.I.C" in: *Proceedings of the First International Conference on Live Coding*. Leeds, p. 220.