Space as Time. A Study in Improvisational Interactive Computational Sculpting

EVAN S. RASKOB

(University of the Arts, London)

Abstract: This paper reviews some of my work in experience-based computational sculpture, using a technique which I call Interactive 3D Printing, an amalgamation of generative art, livecoding, and sculpture. I3DP draws on a rich history of iterative revision and aesthetic refinement in the computational arts. This work foregrounds the time-based experience of digitally fabricating objects by describing them using only terms for time and rhythm (beats, beats-per-minute, duration, musical notes) following Paul Klee's observation that "space itself is a temporal concept". It explores the liminal state between finished/unfinished objects inside a manufacturing process by incorporating the sound of the manufacturing process into the experience of its products. I discuss how this work can be understood as both an improvised livecoding performance and a work of generative art where each iteration (or "instantiation") has the potential to self-actualise and change over time according to the intrinsic nature of both computational and improvisational works of art.

Keywords: Interactive Programming; 3D Printing; Performance; Computational Sculpture; Generative Art.

1. *Introduction*

This paper reviews some of my work in experience-based computational sculpture, using a technique which I call Interactive 3D Printing, an amalgamation of generative art, interactive programming, livecoding, and sculpting. I3DP draws on a rich history of work on interactive, iterative revision and aesthetic refinement in the computational arts. Briefly, this interactive way of working with computational systems runs in contrast to a more common mode of computer programming where a programmer writes an entire computer program code and then *executes* or *runs* it, meaning that it is translated into machine language on some level (i.e. *compiled*) and then put into motion by a computer of some kind. On many systems, once written and executed the program is in effect "dead" to the programmer and cannot be modified or often even interrogated to understand what it is doing (Rusher 2022).

Interactive programming (IP), in contrast, is a more performative and improvisational way of working where a person writes code in parts, incrementally, and then runs just those parts, often whilst other code is running at the same time (Tanimoto 2013; Rein et al. 2019). IP systems often include ways of inspecting, querying, and modifying parts of running code. The argument for such iterative, interactive, and almost conversational ways of working with computers can be seen in early writings by Turing (Parisi 2021). As Jack Rusher (2022) observed, it is built into a few scalable and concurrent programming systems like *Erlang* that expect to be running mission-critical applications without failure for long amounts of time, like cellular phone networks, and cannot simply stop and restart.

To over-generalise, the main difference is that the "normal" mode of programming is focused on writing complete programs with predictable and replicable effects, whereas interactive programming is more interested in the *experience* of programming and its effect on an already-running system. A session of interactive programming may end with the production of a working, self-contained program, but often it can be an end in itself: a way of passing time for the programmer; a learning experience; a scratch pad for sketching out new ideas; or even, in the case of livecoding, an audiovisual or choreographic performance that involves a live audience in the creation of the work (Rein et al. 2019; McLean and Wiggins 2010; Parisi 2021).

With I3DP, where a 3D printer is brought into the interactive programming process, joining process, producer, and product. This entangles the artist, machine, algorithms, and their potential audience in the creation and re-creation of the work, questioning the finality and determinacy of controlled fabrication processes. After all, algorithms and machines are step-by-step processes whose continual outputs depend on their conditions and context: «Instead of imitating the world, algorithms act in the world and can only give us incomplete pictures of a world *in the making*» (Parisi 2021).

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

Figure 1 The sculpture *Indeterminate M* composed of 4x4 beats at 60 bpm version. © Evan Raskob.

It celebrates the complexity of algorithms in motion that lead to surprising, indeterminate outcomes, each outcome a fragment of a wider possibility space that audiences can learn to recognise. Such computational, generative, and rule-based works embody «the unlimited potential that every numerical bit of a program, or every experiential bit of a dance (every gesture and step), has to change and be something else» (Fazi 2018: 39).

The works I introduce in this paper are computational in nature and as such are based on algorithms, or «algorithmic thought» (Parisi 2021). They are improvisational performances using software and digital machines, developed over successive sessions and recorded in physical sculpture and in video. Each work is computational and digital but also physical and bodied, not just in the sense of myself and the machine producing the art but also in the sense of the mixed-media artefacts produced. Their outputs can be considered to be conceptual fragments of computational thought; shards of unpredictable digital processes that have their own explicit rhythm and musicality, explicitly recorded in code written using the special LivePrinter syntax.

My I3DP work foregrounds the time-based experience of digitally fabricating objects by subverting the language we use to describe them, focusing on the parameterised, discreteness of time in a computational sense in juxtaposition with the continuous human time that fills the experience of the artist and audience during the performance-of-making. Instead of the more common practice of describing shapes using precise measurements of space (length, depth, height) I use only terms for time and rhythm (beats, beats-per-

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

minute, duration, musical notes). This severely limits the composer to working with the rhythm and discretised time inherent in the underlying algorithms and fabrications process, called «algorhythmics» by Miyazaki (Miyazaki 2012). How this special time-based syntax is written, and how it turns musical notation into sound using the 3D printer's mechanical system is described in some general detail later on.

In a more visual and experiential sense, this concept builds on Paul Klee's observation that «space itself is a temporal concept» (Klee 1961: 78), the idea that images can convey a sense of time, velocity, and rhythm based on their compositional components like lines and textures. It is inherent in many other works of visual art that are not strictly performative or time-based, like Yayoi Kusama's obsessively repetitive drawings, where the sheer amount of time it took her to finish these detailed works is part of the experience of appreciating them. In music, space is associated with reverberation and echo: the larger the space, the more time it takes for sound to travel across it and reflect back, leading to feedback and overlapping rhythms.

In my work, a compositional substitution of space, usually specified in units of length, with time, usually specified in "notes", helps to account for the *lost time* of manufacturing between when the algorithms codified in virtual forms are set into motion by the 3D printer at the start of the fabrication process, and when the finished physical products appear from the printer at the end. In doing so, it also highlights the liminal state between finished/unfinished objects inside a manufacturing process as an aesthetic experience incorporating the sound of that process into all of its constituent components.

The musical rhythm of machines has a history of influencing musicians and artists but has rarely been used in such a way that intentionally produces both physical and musical outcomes. The early works discussed in this paper begin to explore this new aesthetic of performative computation enabled by I3DP and defined by an amalgamation of algorithms, movement, sound, time, and physicality.

2. *Towards an aesthetics of computational performance*

«Every work of art is both an interpretation and performance of it» (Eco 1989: 4).

Like Mandelbrot's simple fractal equations that feed back into themselves to generate infinitely long surfaces that resemble the jagged coastlines of beaches and islands, this brief statement recursively frames a work of art as a continuing series of ever-evolving experiences, each fed back into itself to

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

become something new. It firmly entangles both the artist and audience in the creation and re-creation of the work, leading to another recursion: «the producer is at the same time the product» (Bertinetto 2013). Some people liken this to the biological concept of «autopoiesis», whereby living things self-produce or self-actualise (Maturana and Varela 1980; Fischer-Lichte 2008).

In computational art practices, the link between the interpretation and performance of a work is explicitly codified in the software system that must interpret the work's computer code and "perform" it through a physical computational system in order to get some kind of output. This type of art draws on a history of machines that embody the artist's creative intentions, whether they are explicitly computational as with software systems, or more implicit in their patterned output, as with weaving machines.

It may seem non-intuitive that a computational work, once set in motion by the artist, can evolve over time, especially since software is often considered predictable and mechanical in how it repeats the same code to a similar effect in what appears to be a deterministic way. Yet, even simple mechanical systems are capable of changing over time and producing surprising outputs, as with the self-destructing mobile sculptures or unpredictable spring-operated *Méta-Matic* drawing machines of Jean Tinguely.

The key is to recognise the physical realisation of these abstract, computational parts of the work because

> algorithms are mathematical and thus abstract structures [...] [they] should not be mistaken for algebraic formulae, since assignments or instructions operated by algorithms are non-reversible. They are vector-dependent and have built-in time functions. Their ties to making reality and operability make algorithms time-based and as such part of rhythmic procedures, which are able to cause measurable temporal effects. (Miyazaki 2012: 1)

Indeterminacy is often inherent in complex computation that relies on the ordering of computation steps in time as (e.g. genetic algorithms and many cellular automata). They are also unpredictable because of the physical nature of their computing machinery (e.g. cosmic ray interference, hardware ageing and obsolescence). These two observations led Wolfram to his *Principle of Computational Equivalence*, which places an upper limit on how sophisticated computational systems can be (Wolfram 2023). That includes human minds, which are bound by Wolfram's Principle to be only as sophisticated as any other "universal" computational system, like rule-based cellular automata

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

systems that he articulated. In other words, for us to understand another sophisticated computational system, we must *perform* the computational process embedded in it by *imagining* it – *running* the rules in our minds. We must couple ourselves to the computer and become «symbolically-ordered» and «in the loop» with the machines (Miyazaki 2012; Ernst 2021: 23)[1].

M. Beatrice Fazi posits that computational artefacts might even have some form of agency of their own, outside human existence, since «computation is not a mere epistemic reduction […] computational processes are dynamic and generative because they have the potential to actualise themselves». Ernst, following Charles Babbage's concept of an «analytic engine», points out that once computation is implemented as physical, «technically-informed» matter, it gains a certain non-human agency (Ernst 2021: 19). These views provide a way of thinking about how to experience and re-experience contemporary computational artistic practices like generative art, and to guide the evolution of new art practices such as generative 3D printed sculpture.

In a way, this brings us back to the autopoiesis of works of art, which have a way of self-evolving after their creation, as some have argued (Iacobone 2021).

> [I]f we somehow get the information that a performance we did not know that was improvised was indeed improvised, then we can retrospectively re-shape the meaning of some of its aspects and the aesthetic merit thereof, because the knowledge of the fact that it has been improvised lets us interpret it in a different way. (Bertinetto 2013: 19)

This conceptual framework helps us understand each iteration of the work, which I will call its "instantiation" in a particular setting, with an audience and human or semi-automated performer(s), as related to a complex combination of its source code, physical (and virtual) hardware it runs on, its creators, collaborators, audience, and the networks it belongs to. The 3D printed sculptures described in this article can be seen as embodiments of the relationship between the performer and printer and the constraints and possibilities of the programming languages used in the performance. Their form and the shape of the performance may be guessed at, but will not be clear to anyone involved in it until they are finished, and even after that they are there

[1] LiveCoder Marjie Baalman creates performances that embody this relationship, such as "Code LiveCode Live" as described in (Baalman 2015).

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

to be seen touched, and reflected upon. This process has no real beginning and no real end – who is to say when the idea first took hold in the mind of the performer? Who can say when the work is truly finished, since each 3D printed work can be replicated, to some convincing degree, and the printer itself can be modified, along with the code?

In this sense, the term "iteration" can be limiting, implying that a process has ended at a particular point and can begin anew, whereas "instantiation" implies a particular instant in an ever-evolving work. Similarly, Ernst uses a term *technológos* (operative diagrammatics) as a shorthand for how «the temporal qualities of algorithms are inevitably related to material structures» (Ernst 2021: 1) but the verb "to instantiate" is more common in programming use to refer to the more tangible entities that allow programmer to interact with an abstract computational process.

3. *Making algorhythmic computational art using 3D printers*

This discussion of theory brings us to the pragmatic question of implementation, since I am writing about works that exist as code, artefacts, and records of performances.

All the works described in this paper were made using the LivePrinter system. LivePrinter is an Interactive 3D Printing (I3DP) system, the result of my practice-based research into how interactive programming can be used to further develop my professional practices of education, computational art and design, and livecoding performance[2]. I3DP is a multidisciplinary technique, linking software engineering, interaction design, artistic practice, product design, materials science and engineering. When it is used in the practice of livecoding, its transparency of intention and outcome can take on a socio-political dimension of radical openness as well.

In livecoding performances, the programmer and their work are often placed at centre stage, with overlapping visuals and graphical coding interfaces visible to the audience instead of keeping them hidden to focus on a singularly seamless and "transparent" user experience. Livecoding's philosophy follows a modern artistic and political tradition of taking an inward-facing, intellectual act that is usually hidden from public view (in this case programming) and turning it outwards as a form of public performance (Blackwell et al. 2014; Roberts, Wakefield 2018). Thus, the values that livecoding embodies are functionally, socially, and aesthetically realised in live performance. It differs

[2] A fuller description of the system can be found in my PhD thesis (Raskob 2022).

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

from many art and design practices that foreground the authorship of the designer/artist in that it acts to re-empower an anonymous programmer whose role is usually obscured from view and whose name is often unknown in commercial software production processes.

I3DP, as a technique, has a more pragmatic lineage. It likely began with mid-20th century computational artists and musicians who created their own interactive tools for making art (Mathews 1963; Franke 1971). During this early era of computing one of the most basic techniques for interactive programming, the Read-Evaluate-Print-Loop (REPL), was first developed. The traditional REPL supported an IP workflow where lines of code are entered, compiled, and then executed as soon as possible, with results made visible (e.g. "printed") on the screen. This empowered a programmer to edit, extend, or otherwise change a running program and experience the result almost immediately afterwards (Tanimoto 2013).

Artists' IP tools help them understand the effects of code so that they might benefit from "minimizing the latency between a programming action and seeing its effect on program execution" (Tanimoto 2013: 31). It allows the computational artist to use the process of "abductive reasoning" (Peirce 1934; Fann 1970) where they experiment based on inference, a mixture of experience and intuition, to quickly iterate through a number of outcomes. The act of programming can be thought of as a running dialogue with a computational system, like LivePrinter, where the artist writes short statements for the system to carry out and then reflects on the results to understand their implications[3].

In her 1975 essay in Leonardo titled "Towards Aesthetic Guidelines for Paintings with the Aid of a Computer", artist Vera Molnar described using such an interactive, iterative process of tweaking code and viewing the results on her computer monitor. This form of "dialogue" was essential to Molnar's graphical form-finding process (Molnar 1975) and to the development of early functional languages like LISP in the same way that interactively programming 3D printer movements using LivePrinter is now a part of my own form-finding, pedagogical and music-making processes. In the case of LivePrinter, this dialogue is carried out in the language of JavaScript mixed with a special shorthand syntax called the "minigrammar". These are compiled by the system into GCode, a near-universal, low-level language for specifying Computer Numerical Controlled (CNC) machine operations like heating the printer,

[3] Here "conversational" is used in both the literal and metaphorical sense – Molnar meant it metaphorically, but in LivePrinter and other interactive text editors there are records of the previous code, leaving literal transcriptions of this conversation between programmer and system.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

moving parts, and extruding plastic. The result of that dialogue is captured by the system and recorded as transcriptions of the conversation between the different parts of the system: between human and LivePrinter, LivePrinter and 3D printer.

3.1. *The conceptual constraints of LivePrinter*

3D printing is a new technique that allows us to fabricate new forms that were previously too difficult or simply not possible to make. Focusing on what is possible with the machines can be overwhelming, so the focus on the constraints of the fabrication process, or what is not possible to do with the machines, can be more creatively useful. Conceptual boundaries like constraints limit our creative possibilities but also focus our inquiry and create clear paths and patterns to follow (Boden 1990: 95).

Conceptual boundaries, which might be called "systems", are very commonly used in art, architecture, and design. A few notable examples: Christopher Alexander's seminal 1966 work "Notes on the synthesis of form" explored patterns in architecture and design; Corbusier's "Le Modulo" system used human proportions (somewhat) to build a theory or architectural aesthetics and utility; artists like Bridget Riley who «set [her]-self limitations, to invent, so to speak, [her] own sonnet form» (Kudielka 1982: 32) in her highly-structured abstract compositions; Freider Nake and his repetitive, computer-driven drawings using «systems art» (Nake 2015); and the artist and Bauhaus educator Paul Klee, who famously developed his *Gestaltung* or theory of form consisting of a set of techniques and concepts leading to «the paths to form, rather than the form itself» (Klee 1961: 17).

The LivePrinter system addresses this problem of the overwhelming possibility of forms by focusing on the movements of the print head, or the "tool path", as it extrudes plastic into space to build forms. It limits artists to three simple constraints:

1. All machine operations must be described using code in the LivePrinter system
2. The user is responsible for specifying tool paths and machine properties (movement, speed, temperature)
3. The machine does all the making, but still with the possibility of human manipulation during that process

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

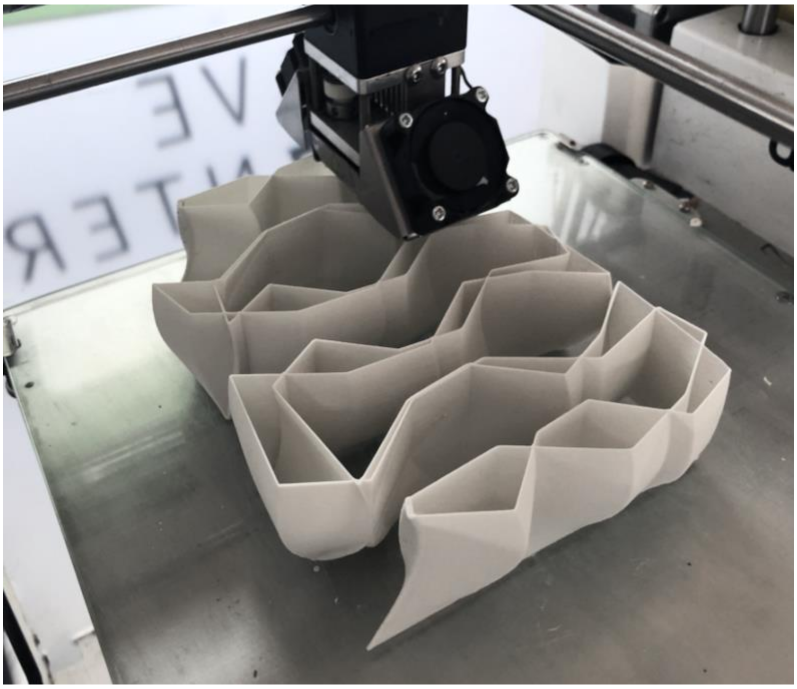

Figure 2 The sculpture titled *Indeterminate W* consisting of 6x6 beats at 60 bpm.
© Evan Raskob.

The constraint of forcing the artist to think about tool paths and other low-level implementation details was at odds with the mostly higher-level geometric and functional abstractions of conventional 3D design software. By bypassing the more familiar language of design in favour of textual metaphors of manufacturing, LivePrinter effectively forces artists and audiences to think about them from strange and different points of view.

Instead of discussing fully-realised forms that could be manufactured, they have to focus on describing the molten plastic lines that incrementally build up such shapes under the pull of gravity and the influence of complex fluid dynamics. Interactive programming becomes a method of defamiliarisation for breaking down the "magic" idea of objects that appear fully-realised from a printer, and reframing it as intentional, detail-oriented, incremental making. This emphasises more the process and time spent making objects, rather than focusing on the outcomes. Focusing on the lived experience of the performer (and audience), working with the dancer-like movements of the 3D printer, and exploring the infinite possibilities afforded by generative algorithms coupled with intuitive, on-the-fly decisions is difficult and unfamilair but also provides a potentially interesting path forwards into creating new sculptural and performance art.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

3.2 *Understanding "algorhythmic" 3D printing*

To understand how a 3D printer's[4] form-making process can be intentionally used to make sounds and compose music, we first need to understand how exactly 3D printers "fill up space" to create free-standing, solid structures from molten plastic. It is also helpful to explain in technical detail how exactly the LivePrinter system supports this form of structural and musical expression.

Typical speed scale for x, y, z axis values for the motors used in the Ultimaker 2 printers to convert their speed into musical notes. From Westcott's MIDI-TO-CNC library (Westcott 2015). Note that no values were given for the filament feeding (e-axis) motor.

X axis	Y axis	Z axis
47.069852	47.069852	160.0

The 3D printer is a mechanical, often box-shaped object. It has a moving print head, which is essentially a heated nozzle that extrudes plastic out its tip to form shapes on a flat surface below, called the print bed. Printers utilise 4 digitally-controlled motors for this task that can be operated individually. The motors move the print head side-to-side (x-direction); forwards—backwards (y-direction); the print bed up—down (z-direction); and feed forward or retract the plastic filament (e-direction). Often, they are the same model of motor and have almost identical mechanical properties.

When the motors of my printer are operating at speed they emit sounds that can be roughly mapped to the musical notes in the equal temperament scale commonly used by MIDI synthesizers using some simple linear scaling in the following JavaScript-like pseudocode (this varies with motor models):

```
// calculate the frequency of the note from
// MIDI note number:

frequency = Math.pow(2.0, (note - 69) / 12.0) * 440.0

// convert to motor speed in millimetres
// per second for GCode (see Table 1)
```

[4] For the purposes of this paper we refer the type that melts plastic filaments in a process commonly called Fused Deposition Modelling (FDM), or sometimes Fused Filament Deposition (FFD) (Livesu et al. 2017; Hoskins 2014; Gao et al. 2015).

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

```
speed = frequency / speed_scale_for_axis
```

In this example, `speed_scale_for_axis` is a simple fraction determined experimentally.

Knowing the travel speed of each motor that produces a desired musical note, along with the desired duration of that note, one can calculate the distance of travel across each axis by using a simple movement equation:

```
d = st
```

where d is the distance in mm to be calculated, s is a scalar representing the speed of the print head in mm/s in the current direction of travel, and t is the desired movement time in seconds. The first, called `_midi2speed(NOTE)` or shortened to `m2s`, converts a MIDI note `NOTE` into a corresponding motor speed in mm/s, for one axis. Then, a function called `_time2dist(TIME)` or the shorter `t2d` can be used to convert that speed and a desired duration of movement_ `TIME` into a movement distance[5].

The following pseudocode uses these two functions to move the print head making a pitch of MIDI note C5 with a duration of 6000 milliseconds:

```
# midi2speed "C6" | drawtime 6000
```

This code will move the print head a distance of 133.3978 mm at a speed of 22.2330 mm/s.

The length of movements can also be specified in terms of beats at a particular rhythm, or “bpm” (beats-per-minute). In this example, we set the printer movement speed to a midi note C5 and move 1 beat in the current direction at 120 bpm:

```
# midi2speed "C6" | bpm 120 | drawtime "1b"
```

[5] There is a caveat to this simple distance function – 3D printer motors do not have unlimited torque, and so they take a brief but perceptible time to accelerate to full speed. That means that, in practice, movement durations are lengthened as the movement speed increases. In our experiments, this was perceptible around MIDI notes 81 and above and caused synchronisation issues when we tried to pair the printer with other musical equipment. The acceleration curve for movements depends heavily on the mechanics of the printer, the type of motor, the motor driver, and any firmware-level acceleration settings, so the exact amount of lag would need to take all that into account.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

This moves the print head 11.1165 mm. We can calculate this using another helpful bit of code, `# t2mm("1b")` or "time-to-millimetres" which returns the length of a movement in millimetres. This was helpful in determining the dimensions of some sculptures, especially the height of some. Using these simple methods, we can use only time-, music- and rhythm-based notation to specify the movements of 3D printers and have no need for other dimensions.

Once the printer is set in motion there is still more work to be done to record the work in video and audio, involving cameras, contact microphones attached to the printer motors, and various audio mixing and signal rectifying devices. It is a formidable technical setup.

4. *Performing space with time and rhythm*

The sculptures presented here are based on a very simple conceptual constraint: to use a grid as a start and end point for crossing all points in space across a rectangular plane, keeping a steady rhythm of points (in beats-per-minute), whilst transitioning between order (rigid structure) and chaos (unpredictable noise) in both space and sound.

This compositional approach also plays on Eco's take on Shannon's

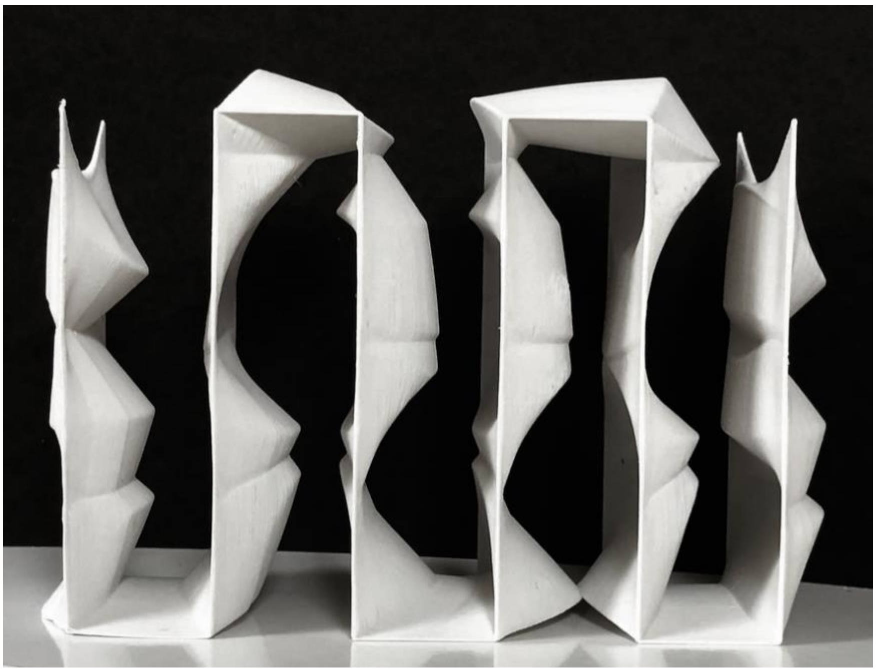

Figure 3 The sculpture titled Indeterminate W consisting of 6x6 beats at 60 bpm placed in a grid whose points shift over time and are also spatially manipulated during the performance. © Evan Raskob.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

Information Theory (Shannon 1948) applied to the concept of "meaning" in art, where "meaning" comes from a set of regular rules that the artwork establishes and then violates to create a sense of novelty:

> From the point of view of communication, I have information when (1) I have been able to establish an order (that is, a code) as a system of probability within an original disorder; and when (2) within this new system, I introduce–through the elaboration of a message that violates the rules of the code–elements of disorder in dialectical tension with the order that supports them (the message challenges the code). […] the disorder that aims at communication is a disorder only in relation to a previous order. (Eco 1989: 58)

These works also take as reference Vera Molnar's earlier works that worked with grid-like, constrained geometries to play with the effects of different "noisy" algorithms and techniques. The interplay between noise and visual perception and their effect on the aesthetics of a work was one that V. Molnar and F. Molnar wrote about often throughout their careers (Molnar, Molnar 1989). Similarly, in Andy Lomas's work he finds interesting forms "at the boundary between regularity and chaos", as visualised in his 2-dimensional tables comparing the effects of a limited number of parameters against one another (Lomas 2018). The amount that a foreground "signal" (e.g. a deterministic spatial algorithm) can be "buried" in a noisy background (e.g. indeterminate results) can be seen as related to Eco and Shannon's formulations of "information" and "meaning", respectively.

In my works, the "established order" or "code" is the regular zigzag grid, defined by a number of points specific to each work. Each piece begins with the grid, or code, and ends with it. In between is a deliberate "chaos" shifting the underlying points in the grid with each successive layer of plastic, that is not random but irregular enough to still convey some sense of deeper order the viewer, and to myself. The functions that create this semi-disorder are themselves time-based, the filling in a sort of chaos-disorder-chaos sandwich. Also, looking across the work from left to right, or top to bottom, there are complex symmetries, hinting at a deeper order.

The hope is that the audience recognises the experiential rules or "code" of these works in the spatial grid and regular rhythm of the machine's movements and then experiences a sense of ambiguity and creative tension through the repeated violation of that code, as the printer traverses the shifting grid points in space at a constant rhythm whilst the outline of that shape and

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

sound change over the duration of the performance. At the end, the performer and audience are left with a trace of the experience of making the object for others to follow when they re-experience it, in addition to the audiovisual recording of the making of the piece, which is also an essential part of the work.

The visual, physical and sonic rhythms as key to understanding the algorithms that created the work in both time and space. The meaning of the works and their sound are open to retrospective re-shaping, because the knowledge of how they were improvisationally made allows future audiences to re-imagine this process in their own way, and re-perform that act in their own minds (Bertinetto 2013).

5. *Reflecting on the performative experience of making*

As an example of this process, in the work *Indeterminate M*, or "60 bpm, 4x4 beat object number 1" (**fig. 1**), I used the LivePrinter system to create an algorithm that generates a 4x4 grid of points where each point is exactly one "beat" away from each other when traversed using the "zigzag" approach (up a column, across a row, down a column, repeat). In this case, the "beat" is a variable that changes with each composition, but is defined in terms of a "bpm" (as described above) such as 60 bpm, and as some multiple or fraction of a beat, e.g. "1/2b" for 1/2 of a beat. This creates a basic rhythm for the work in time, space, and sound.

The shape is constructed by starting with a rectilinear grid where all movements use only a single motor (in either the x or y direction) and thus produce more "pure" music tones, and then, as each layer is extruded and the height increases, smoothly shifting these points until they are diagonal to one

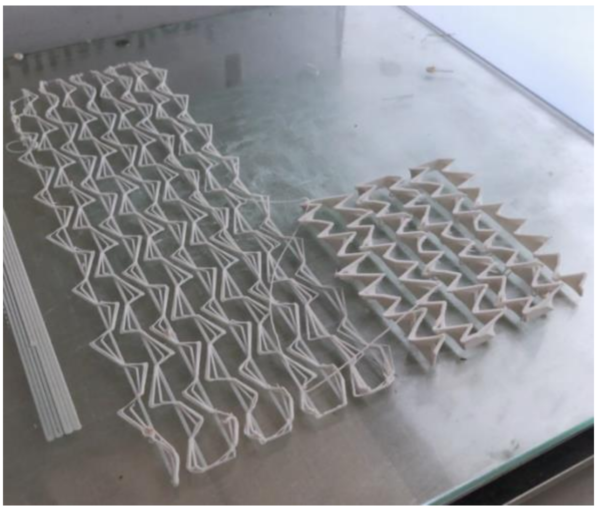

Figure 4 Some unused trials of longer grids (9x88 and 6x22 beats). © Evan Raskob.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

another so that the movements between them use different ratios of each motor in concert, producing complex chords that drift in pitch over time yet stay at the same duration of a single beat.

By using just a few points, and keeping the printing speed relatively slow and even, this creates a rhythmic droning sound that resembles a stringed instrument being amplified. Neither sound nor shape are too complex to interfere much with the experience of one another, whilst also being irregular and unpredictable enough to keep the experience interesting but not wildly chaotic. In the end object, the visual and sonic grid form is recognisable at the extents of the object, but not overwhelmingly so.

Some discarded trials used more beats per grid to produce more active and energetic sounds, but I wasn't happy with the physical aesthetics. These violated the rules of the regular grid so often that their transgressions became regular and thus less meaningful and exciting, in Eco's formulation. I am reminded of Molnar's description of some of her discarded drafts in her *Hypertransforms* series as "disappointing", and like her, I find the feeling more intuitive and experiential than readily explainable (V. Molnar 1975), in spite of the conceptual formulations of order vs. chaos and noise vs. signal.

The more I work with these sculptures as performances, the more I come to appreciate both the wholeness of the performance itself, which requires constant unbroken presence in the sense of "La Dureé" – the longest work, "Indeterminate W" (**fig. 3**), took a full 2 hours, 48 minutes and 49 seconds to fully resolve when printing at a fairly slow and even rate, creating an ambient soundscape that backgrounded my other work. It is within the liminal states of construction – after the start of the physical making of the object but before the form is closed – where the logic of its construction is revealed.

Inside this liminal space I accidentally discovered that a mistake in one of my algorithms led to Lissajous-like forms where out-of-sync sinusoidal forms crossed over one another instead of following the same paths in a slowly shifting manner, as seen in **fig. 2**. In these forms, the algorithm for subsequent layers became out-of-sync with each beat position. The crossing-of-paths is usually avoided in 3D printing because the moving print head may break the shape when it hits it on the build plate, but in this case the intersection points were too small to make a difference and I was able to compensate for the subsequent gaps between successive layers, resulting in fully-formed shapes that were more intentionally developed over subsequent iterations and performances.

In these particular forms, which evolved to the works shown in this paper (**fig. 1**, **fig. 3**), the overlapping spatial and temporal rhythms of these forms becomes smoothed over in the finished, enclosed shape, locking away evidence

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

of the transgressions of the errant algorithms. I found the evidence of the "mistakes" interesting, and was glad that the videos captured their construction. This led to more interactive experiments exploring where the outcome of the work became unpredictable – I might have decided to nudge the sculpture towards completion by changing the bpm of the piece; the software may have become unstable for a minute, leading to a speed change and a momentary under-extrusion; I may have missed a cue and changed a parameter slightly off-beat; or the temperature changed, and a layer didn't quite stick.

The resulting objects bear the traces of all of these changes in its bumps, grooves, and "glitches". I feel Barad's sense of a "dis/continuity" in them (Barad 2013), and in the intermediate sounds of the piece as the motors change direction and they transition from musical monotone to richer chords, moving from chromatic order to discordant dissonance. There is an order that changes, sometimes abruptly, but it is a turning from an established path and not a reversal or a break with that past.

What is clear about making the pieces is the messiness of the *entanglement* of the work with audience, performer, machine, and code, in Barad's sense of the word (Barad 2013). I find myself dreaming of code sometimes whilst looking at the finished or discarded or aborted shapes, imagining what might be, deconstructing the finished objects in my imagination to explore other algorithmic approaches, wondering what other transgressions are structurally and performatively possible.

5.1. *Drawbacks and future opportunities*

There are a few areas that need further developing, both critically and technically, since these are tightly intertwined. One major area is a lack of visual representations of the code used to move the 3D printer. Concentrating on only the textual algorithms that generate the forms means limiting the range of algorithmic thought available to the artist, making it difficult to think purely spatially (for example, in using arcs and trees) without the difficult mental operations of translating them to textual descriptions.

This also implies a lack of structured interaction with some intermediate data structures created by the drawing algorithms. For example, once a form is described in code and compiled, it can only be manipulated by special, named variables that were explicitly included in the code. This requires a measure of forethought and planning that complicates free improvisation.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

Both of these issues are open areas of research in computing, but also not easily addressed without switching from the artistic, performative mode to a deeper research and development mode focused primarily on tool-building. While that would be an interesting research topic, it highlights the difficulty and relatively slow pace of tool-building and software development for artistic purposes by artist-developers who create their own systems.

A different issue is the difficulty of presenting fragmented works to others. As the performer/artist/developer, it is a difficult task to step out outside those roles and choose particular artefacts, moments, sounds, code examples, and other selections from works of long durations. This is a common problem of generative and computer-aided art and design: what is part of the exhibited work, and what is part of the explanation of it. With generative works that can produce infinite numbers of variations, the problem become more acute and near paralysing.

6. *Conclusion*

The limited number of works presented here illustrates some new ways of working with interactive programming that explore the fragmented nature of computational thought. These early works explore ways in which this new aesthetic of performative computation (movement, time, and physicality) might emerge. They exist simultaneously as fragments in different mediums, of code, physical form, video, experience, and sound art. The fact that they are tied to concrete forms created from a physical experience makes them special.

Each "instantiation" of the work at different points in space-time lets observers experience a different combination of fragments drawn from this larger, continuous computational/physical space. Also, as physical computational works they can be seen to be continuously self-actualising, both in the minds of current and future observers and in the wider world as a result of recursively combining social, artistic, and computational processes.

This view also challenges notions of finality in a work, especially generative artworks. There is a common meme on social media where people joke about how the current filename for their work is something like "FINAL FINAL v3 FINAL (COPY)", referencing all the so-called "final" versions of their work that came before but were intermediate iterations that may never lead to a truly final result. That is why I use the term "instantiation" to loosely refer to a specific form of the work that exists in space and time. This term, with its sense of a multiplicity of possibilities, rejects the idea that works can be finished to some "final" conclusion but also embraces the idea that they take

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

on definite forms at a specific time and for specific purposes, like at an art installation, an online auction, or during a live performance.

This is not an uncommon or revolutionary view, but it is important to understand when experiencing my work and the work of other generative artists. It also forms the main dilemma of such works, which is how to best share them with others? That process involves composition, creation, curation and play, where the artist works on the computational processes that generate the work and then decides on the level of curatorial responsibility they would like to assert over the results and the ways in which the work physically or virtually manifests to its audience. This framing reminds us that the possibilities inherent in the work, and how much control is granted over those possibilities to participants to play with and imagine is at the heart of the artist's intentions for the work.

The hope is to make works of art that are conceptual and reflective, but also surprising and interesting as audio-visual-tactile experiences. The exposed process of coding and fabricating this art goes against Frieder Nake's view that "there is no emotion" in the making of generative art (Nake 2015, time: 19:49) and the process of form-finding is necessarily, as he puts it, «boring as hell». It also contains a subtle critique of the need we have to make every process more *efficient* – why not make them more *interesting* instead? What are we saving by trading one experience for another, especially in the creation of art?

evan.raskob@arts.ac.uk

Bibliography

Baalman, Marije (2015). Embodiment of Code. *Proceedings of the First International Conference on Live Coding*, 35-40. Leeds, UK: ICSRiM, University of Leeds (https://doi.org/10.5281/ZENODO.18748).

Barad, Karen (2013). Ma(r)king Time: Material Entanglements and Re-memberings: Cutting Together-Apart1. In P.R. Carlile, D. Nicolini, A. Langley, H. Tsoukas (eds.), *How Matter Matters: Objects, Artifacts, and Materiality in Organization Studies*, 16-31. Oxford: Oxford University Press.

Bertinetto, Alessandro (2013). What Do We Know Through Improvisation? *Disturbis* 14, 1-22 (https://ddd.uab.cat/record/123756).

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

Blackwell, Alan, McLean, Alex, Noble, James, Rohrhuber, Julian (2014). Collaboration and learning through live coding (Dagstuhl Seminar 13382). In A. Blackwell, A. McLean, J. Noble, J. Rohrhuber (eds.), *Dagstuhl Reports* 3(9), 130-68 (https://doi.org/10.4230/DagRep.3.9.130).

Boden, Margaret A. (1990). *The Creative Mind: Myths and Mechanisms*. London: Weidenfeld and Nicolson.

Eco, Umberto (1989). *The Open Work*. Cambridge, Mass.: Harvard University Press.

Ernst, Wolfgang (2021). Existing in Discrete States: On the Techno-Aesthetics of Algorithmic Being-in-Time. *Theory, Culture & Society* 38(7-8), 13-31.

Fann, K. T. (1970). *Peirce's Theory of Abduction*. 3Island Press.

Fazi, M. Beatrice (2018). *Contingent Computation: Abstraction, Experience, and Indeterminacy in Computational Aesthetics*. Media Philosophy. London, United Kingdom; Lanham, Maryland: Rowman & Littlefield International.

Fischer-Lichte, Erika (2008). *The Transformative Power of Performance. A New Aesthetics*. London: Routledge.

Franke, H. (1971). Computers and Visual Art. *Leonardo* 4(4), 331–38.

Gao, Wei, Zhang, Yunbo, Ramanujan, Devarajan, Ramani, Karthik, Chen, Yong, Williams, Christopher B., Wang, Charlie C. L., Shin, Yung C., Zhang, Song, Zavattieri, Pablo D. (2015). The Status, Challenges, and Future of Additive Manufacturing in Engineering. *Computer-Aided Design* 69, 65-89.

Hoskins, S. (2014). *3d Printing for Artists, Designers and Makers: Technology Crossing Art and Industry*. London: Bloomsbury Academic.

Iacobone, Alice (2021). Improvisation in Sculpture. In A. Bertinetto, M. Ruta (eds.), *The Routledge Handbook of Philosophy and Improvisation in the Arts*, 585-99. New York: Routledge.

Klee, Paul (1961). *Notebooks Volume 1: The Thinking Eye*. Ed. by J. Spiller. Transl. by R. Manheim. London: Lund Humphries.

Kudielka, Robert (1982). The Paintings of the Years 1982-1992. In *Bridget Riley, Paintings 1982-1992*. Nuremberg; London: Ex. Cat.

Livesu, Marco, Ellero, Stefano, Martínez, Jonàs, Lefebvre, Sylvain, Attene, Marco (2017). From 3D Models to 3D Prints: An Overview of the Processing Pipeline. *Computer Graphics Forum* 36(2), 537-64.

Lomas, Andy (2018). On Hybrid Creativity. *Arts (Basel)* 7(3), 25.

Mathews, M. V. (1963). The Digital Computer as a Musical Instrument. *Science* 142 (November), 553-57.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

Maturana, Humberto R., Varela, Francisco J. (1980). *Autopoiesis and Cognition: The Realization of the Living*. Boston Studies in the Philosophy of Science; v. 42. Dordrecht: D. Reidel.

McLean, Alex, Wiggins, Geraint A. (2010). Bricolage Programming in the Creative Arts. In *PPIG*, 18.

Miyazaki, Shintaro (2012). Algorithmics: Understanding Micro-Temporality in Computational Cultures. *Computational Culture* 2 (http://computationalculture.net/algorhythmics-understanding-micro-temporality-in-computational-cultures/).

Molnar, F., Molnar, V. (1989). Noise, Form, Art. *Leonardo* 22(1), 15-20.

Molnar, Vera (1975). Toward Aesthetic Guidelines for Paintings with the Aid of a Computer. *Leonardo* 8(3), 185-89.

Nake, Freidrich (2015). Art That Makes Itself Symposium 2015: Frieder Nake. London (https://www.youtube.com/watch?v=ICOs_8pjOIc).

Parisi, Luciana (2021). Interactive Computation and Artificial Epistemologies. *Theory, Culture & Society* 38(7-8), 33-53.

Peirce, Charles S. (1934). *The Collected Papers of Charles Sanders Peirce, Vol. V: Pragmatism and Pragmaticism*. Ed. by C. Hartshorne, P. Weiss. Cambridge: Harvard University Press.

Raskob, Evan Saul (2022). *Working Performatively with Interactive 3D Printing: An Artistic Practice Utilising Interactive Programming for Computational Manufacturing and Livecoding*. PhD thesis, Goldsmiths; Goldsmiths, University of London (https://doi.org/10.25602/GOLD.00032798).

Rein, Patrick, Ramson, Stefan, Lincke, Jens, Hirschfeld, Robert, Pape, Tobias (2019). Exploratory and Live, Programming and Coding: A Literature Study Comparing Perspectives on Liveness. *The Art, Science, and Engineering of Programming* 3(1).

Roberts, Charles, Wakefield, Graham (2018). Tensions and Techniques in Livecoding Performance. In A. McLlean, R. T. Dean (eds.), *The Oxford Handbook of Algorithmic Music*. Oxford: Oxford University Press.

Rusher, Jack (2022). Stop Writing Dead Programs (https://jackrusher.com/strange-loop-2022/).

Shannon, Claude (1948). A Mathematical Theory of Communication. *Bell System Technical Journal* 27(3), 379-423.

Tanimoto, Steven L. (2013). A Perspective on the Evolution of Live Programming. In *Proceedings of the 1st International Workshop on Live Programming*, 31-34. LIVE '13. Piscataway, NJ, USA: IEEE Press.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040

Westcott, Matt (2015). "MIDI-TO-CNC." September 2015 (https://github.com/gasman/MIDI-to-CNC/blob/master/mid2cnc.py).
Wolfram, Stephan (2023). The Principle of Computational Equivalence (https://www.wolframscience.com/nks/chap-12--the-principle-of-computational-equivalence/).

Evan S. Raskob is currently a Senior Lecturer and Course Leader for BSc Computer Science at the Creative Computing Institute at the University of the Arts, London (UAL). He creates computational art using data and interactive programming for 3D printers; develops live architectures for interactively programming 3D printers and creating generative art; consults with small and large businesses for software development, especially on the web; creates science fiction/speculative design card games; composes and performs live music and visuals, often with code (livecoding); develops international workshops on topics like for interactive media, mixed reality, VR, augmented sensory design, and electronics for artists and designers.
In the past Evan has collaborated with the artists Robert Whitman and Julie Martin, both of the E.A.T. collective; helped create an immersive, interactive LED lighting system for the Fairchild Communications (Vogue Magazine, etc.) flagship offices in New York City with architectural firm SOM; and worked closely with companies such as BBH and Barclays on student placements and user experience design consulting projects.

ISSN: 2036-542X
DOI: 10.13135/2036-542X/9040