Sculpting Algorithmic Pattern: Informal and Visuospatial Interaction in Musical Instrument Design

Jack Armitage, Thor Magnusson
Intelligent Instruments Lab
Iceland University of the Arts
Reykjavik, Iceland
jack@lhi.is,
thor.magnusson@lhi.is

Andrew McPherson
Augmented Instruments Lab
Dyson School of Engineering
Imperial College London
London, UK
andrew.mcpherson@imperial.ac.uk

ABSTRACT

In live coding, the concept of algorithmic patterns is employed to characterise the improvisation of artistic structures. This paper presents a digital musical instrument (DMI) design study that led to the development of our understanding of proto-algorithmic thinking. The study focused on tools for manually sculpting digital resonance models using clay, with participants following brief technical instructions. The resulting thought process was grounded in systematic embodied interaction with the clay, giving rise to a form of algorithmic thinking that precedes the conceptual formalisation of the algorithm. We propose the term 'proto-algorithmic pattern' to encompass implicit, tacit, gestural, and embodied practices that lack a formalised notational language. In conclusion, we explore the implications of our findings for interfaces and instruments in live coding and identify potential avenues for future research at the intersection of live coding and DMI design.

1. INTRODUCTION

Live coding systems in the arts enable algorithmic expressivity, providing a notational language for artistic patterns and allowing users to frame their thoughts using the operational symbols of the language. However, algorithmic expression can be considered broader than this definition. In this paper, we present an example of what we describe as *proto-algorithmic patterning*. In a New Interfaces for Musical Expression (NIME) 2020 paper, McLean offered the following reflection and definition of algorithmic pattern:

> "Algorithmic pattern" seems to be a tautology. The phrase is nonetheless useful for on one hand clarifying that we address algorithms not just as software engineering tools, but as formalised ways of making that can to a large extent be perceived in end-results. It also clarifies that we are interested in patterns that are not just simple sequences, but structural qualities. This builds a perspective on pattern as a generative and perceptual connection between creation and reception. In short, I define algorithmic pattern as the perception of systematic activity. [1]

In this study, we aimed to comprehend the outcomes of a digital musical instrument (DMI) design investigation, wherein instrument creators moulded digital resonance models utilising physical clay [2]. As live coders ourselves, we were intrigued by the concept of algorithmic pattern as a means to understand participants' sculpting, due to our perception of systematic activity in their responses. However, we discovered that the existing definition of algorithmic pattern precluded our observations from being classified as such, as the clay sculptures were tacit and informal, devoid of any notational language or possessing the status of discrete symbols. This prompted us to further examine the relationship between our findings and algorithmic pattern, and in doing so, elucidate the tacit aspects of live coding.

The user behaviour we observed clearly indicated algorithmic tendencies, but not in a manner that was entirely familiar or directly comparable to existing live coding practices [3]. Moreover, it was not fully encapsulated by the concept of heuristics in the psychological sense [4], due to the intimate, non-verbal, and continuous material dialogue with the clay and sound. We characterise what we witnessed as *proto-algorithmic pattern*, which we define as any activity where material engagement with symmetry results in the emergence of symbolic and procedural motifs that initiate dialogue between the variant and invariant qualities of experience. The implications of such a framing of operating with DMIs can pave the way for an approach and design focus that enables more fluid, pre- or proto-symbolic, fuzzy interfaces for live coding performance and other interfaces involving algorithmic patterning [5].

In this paper, we explore the boundaries of algorithmic pattern and examine the connections between live coding and DMI design research. We present the interfaces and activities designed for the study, along with a thematic analysis related to algorithmic patterning. We then report on our quest for an understanding of algorithmic pattern that encapsulates our outcome interpretations, and how speculatively broadening our conception of algorithmic pattern became a generative process for rethinking expressive algorithmic interfaces in live coding and more broadly.

2. BACKGROUND

2.1 Algorithmic Pattern and Live Coding Research

To function as symbolic agents in the world, we consistently engage with discrete elements in the form of language, description, notation, and communication in general. This process is what Bernard Stiegler refers to as grammatisation: "By grammatisation, I mean the process whereby the currents and continuities shaping our lives become discrete elements [...] Writing, as the breaking into discrete elements of the flux of speech, is an example of a stage in the process of grammatization" [6]. Our languages provide methods for dividing the world into discrete elements, enabling us to interact with them, memorise, set goals, and exchange instructions. Continuous movements transform into symbolic gestures within the society of hermeneutic agents when repeated or defined as such, often generating tension with our non-verbal, everyday embodied experiences.

Delving further into the concept of algorithmic patterns, an algorithm is described as "a step-by-step set of instructions," while a pattern is defined as "a whole family of techniques for working with regularities in the world. Such patterns enable us to perceive repetition, reflection, and interference in a material" [1]. To fulfil the criteria of an algorithmic pattern, a situation must encompass systematic repetition of motifs, discretised instructions, and a human or agent applying these motifs and instructions to a material.

Moreover, it seems necessary for the actual manipulation to occur through a notation, as in live coding, where "every event arises from notation with explicit, formal meaning [...] code is by nature deterministic" [7]. From this viewpoint, notation serves as an essential expression of experience, an explicit formalisation of thought, resulting in an interface "one step removed" from the material itself [1]. The lexicon of notations consists of symbols, "signs relating to a signless experience" [7], represented by marks that embody expressive linguistic potential.

In programming, these symbols typically take the form of discrete alphanumeric sequences, which can nonetheless be "applied to simulate analogue systems, including aspects of the external analogue world. Furthermore, human cognition [...] involves both analogue and discrete processing" [8]. To date, the predominant focus in live coding research and practice has centred on conceiving algorithms and notations as formal constructs existing outside the human mind. Alongside this focus, we observe a bias towards algebraic formulation and discrete, linguistic manipulation, which is reflected in the etymological proximity of algebra and algorithm. Challenging this status quo is the Hybrid Live Coding Workshop (HLCI) [1] , proposed as a platform for projects and ideas that may not fit within the already expansive International Conference on Live Coding (ICLC). This highlights a community-driven need for more comprehensive and nuanced discussions surrounding the subjective aspects of live coding.

2.2 Informality, Non-notation, and Visuospatial Cognition

The actual implementations of algorithms on computers frequently compromise formalism in various ways due to hardware limitations and performance requirements, such as when dealing with irrational numbers. This implies that, in reality, live coders primarily interact with perceptually close approximations of formality. Moreover, although the original TOPLAP manifesto proclaims that "algorithms are thoughts" [2] , programmers do not strictly adhere to formal algorithmic principles. When combined in the "impossible" act of live coding, where "the relation between code and result is necessarily [technically] nonobvious" [9], any formalisms become part of an open-ended system, in which error and entropy are often embraced.

Live coding, then, is almost or not quite algorithmic, and it is more practical to consider it as navigating a continuum of formality, never reaching chaos or order. In general, the majority of live coding systems appear to be clustered around the more formal end of this continuum, often overlooking the powerful resource of ambiguity [10]. Informal or approximate problem-solving strategies, sometimes known as heuristics, have been investigated by psychologists for many decades [4]. Nevertheless, as Clark points out, heuristics are merely labels for the fast and cheap end of the "continuum of self-organising dynamics", and in reality, embodied organisms tend to "spread the problem-solving load between brain, body, and world" [11].

The Live Coding User's Manual cites Boria *et al.*'s criteria for defining a notation system: "Is there an inner logic?... Is there a vocabulary? ... Are the notations potentially accessible to at least one entity/person?... Are other aspects intentionally left out?" [7]. If a live coder has successfully mastered a notation, it implies that they can utilise it for algorithmic patterning in their mind, even if the resulting pattern may be too intricate to envision. However, open questions persist concerning the actual mental processes involved in tacit algorithmic patterning with notations.

Furthermore, although live coding posits that "creativity involves employing visuospatial cognitive resources to navigate an analogue search space with geometric structure" [8], programming systems have scarcely engaged with visuospatial cognition. Victor, in one of his numerous critiques of visual art tools, emphasises that "code is algebraic, we think linguistically when we're coding, great art has always been created by thinking visually [...] it needs to be grounded in the visual mathematics, which is geometry not algebra" [12]. Eglash states that "a geometric algorithm provides explicit instructions for generating a specific set of spatial patterns" [13]. However, there are currently few live coding systems that have delved into the realm of programming through geometric construction and spatial relationships; "visual programming languages" exhibit minimal actual spatial syntax, employing space primarily as "secondary notation" [8]. We report further on the deficencies of these systems in [14, 15].

---

[1] `https://hybrid-livecode.pubpub.org/`

[2] `https://toplap.org/wiki/ManifestoDraft`

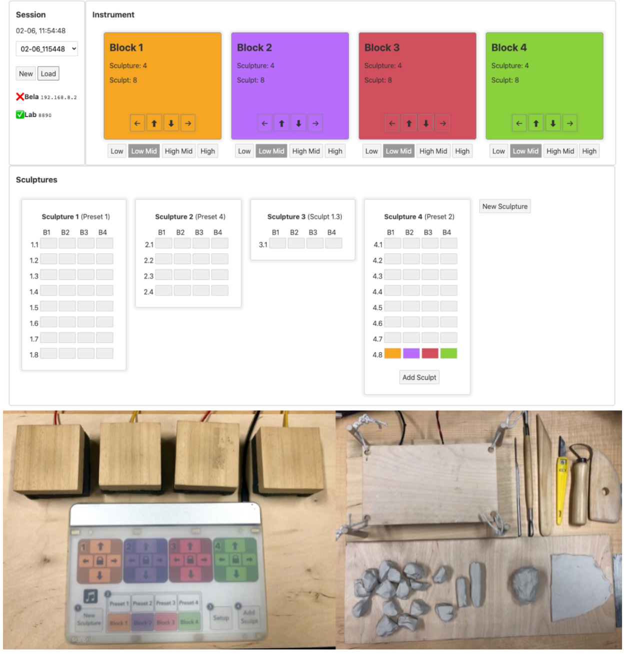

Figure 1: Above: graphical user interface. The Session panel (top left) displays administrative features. The Instrument panel (top) shows the state of the four tuned percussion blocks. The Sculptures panel (below) presents each Sculpture as a table, and each individual Sculpt as a row. Below: photographs of the physical interfaces. Left: digital tuned percussion instrument (above) with four playable blocks, and tangible user interface (below) for navigating the GUI. Right: sculpting surface (above left), sculpting tools (above right), and sculpting clay (below).

3. DIGITAL RESONANCE CLAY SCULPTING SYSTEM

The primary objective of the clay sculpting system presented in this paper was to enable the hand-crafting of digital resonance models through an interface that allows users to make minute, yet consistent adjustments, while concentrating on the intricate nuances of sound. For a complete demonstration of the sculpting workflow, please refer to the video demo[3]. The system was developed as part of a PhD thesis about subtelty and detail in DMI design [2]. The ontology that describes how we define subtlety and detail in DMI design is presented in [16], and an interpretation of this study from that perspective can be found in [17].

3.1 Sculpting Interfaces

The graphical user interface is illustrated in Figure 1 (above), while the physical interfaces are presented from the participant's perspective in Figure 1 (below). The tuned percussion instrument comprised four identical wooden blocks, each equipped with a piezoelectric vibration sensor mounted underneath. The resonance model sculpting tool consisted of a suspended wooden panel with a piezoelectric vibration sensor and a vibration transducer mounted beneath it. A tangible user interface, based on a Sensel Morph[4] device, controlled the sculpting process and facilitated navigation of previous sculptures in conjunction with the graphical user interface. The graphical user interface displayed three panels of information regarding session state, instrument state, and sculpture state.

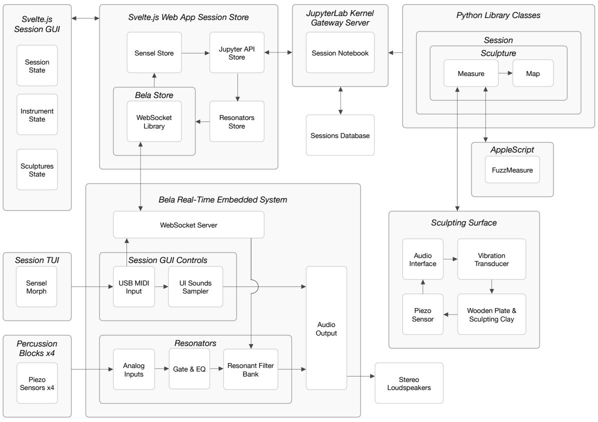

Figure 2: System architecture diagram.

3.2 Sculpting Workflow

Sculpting in this activity referred to utilising the sculpting surface, clay, and tools to manipulate the digital resonance models excited via the tuned percussion blocks. This process was discretised into *Sculptures*, which comprised a sequence of individual *Sculpts*. A *Sculpture* took an existing resonance model, calibrated it to the sculpting surface, and then allowed the participant to manipulate the model based on further changes to the sculpting surface, recorded as *Sculpts*. A *Sculpt* consisted of a frequency response measurement of the sculpting surface and a new resonance model, created by mapping differences in frequency response to model parameters.For every *Sculpt*, the mapping algorithm contrasted the current frequency response with the calibration frequency response, and assigned differences between them to the parameters of the chosen preset resonance model, thereby generating a new resonance model. Hence, each *Sculpture* signified a collection of variations on a particular preset resonance model, connected by the physical *Sculpts* that occurred.

The sculpting workflow comprised four steps, which were illustrated during an introductory tutorial and numbered on the tangible user interface (Figure 1, below):

- *New Sculpture*: this step involved creating a new sculpture and incorporating it into the session.
- *Preset Selection*: the participant was asked to choose a resonance model for the new sculpture from four available presets.

- *Setup*: this step entailed taking a frequency response measurement of the sculpting surface as a reference, which was subsequently utilised by the mapping algorithm to generate new models.
- *Add Sculpt*: this step was carried out iteratively, comparing new frequency response measurements with the *Setup* measurement to map the resonance model.

---

[3] `https://www.youtube.com/watch?v=EtJrk9LywWI`

[4] `https://morph.sensel.com/`

3.3 System Architecture

The system architecture is illustrated in Figure 2. The components of the system encountered by the participant included the Percussion Blocks, Sculpting Surface, Session GUI, and Session TUI. Internally, the tuned percussion instrument operated on a Bela embedded system, while the JupyterLab scientific computing environment managed the sculpting surface sampling and mapping processes. The logic necessary to coordinate interactions between these two subsystems and handle user inputs was executed in a Svelte.js Web App, with a group of Stores facilitating state changes as required.

4. SOUND SCULPTING ACTIVITY

We guided 26 instrument makers and musicians through one hour of brief creative and technical activities, followed by 30 minutes of semi-structured interviews. A comprehensive description of this activity design can be found in the corresponding PhD thesis [2].

4.1 Participants

Participants were recruited through a call for volunteers posted on mailing lists related to digital media arts and traditional musical instrument making. The call stated that participants would “be introduced to a set of digital handcrafting tools and asked to use them to carry out a set of tasks” over a 60-minute activity, followed by a 30-minute discussion. Interested individuals applied to join the study by completing a survey in which they self-reported their experience levels in music, handcraft, and instrument making. A total of 33 participant sessions were conducted, but seven of these were excluded from the final analysis due to methodological consistency reasons, leaving 26 sessions in total. Upon arrival for their session, participants reviewed a proforma consent form containing a description of the activity.

4.2 Activity Protocol and Interview Topics

The primary objective of this activity was to swiftly enable participants to independently manage the sculpting workflow and process, and subsequently immerse them in specific contexts necessitating engagement with the intricate aspects of the materials. The one-hour activity comprised four tasks: Demo (15 mins), Matching Task 1 (5 mins), Tuning Task (20 mins), and Matching Task 2 (5 mins).

The Demo provided a hands-on guided tour of the interface features, including a period for free exploration. The Matching Task, a brief technical exercise, was performed twice to facilitate comparison of results before and after the more extended, creative Tuning Task.

In the Matching Task, participants assumed the role of a sound sculptor who had been commissioned by a musician to create a sound closely resembling another provided sound. During the Tuning Task, participants took on the role of a musician and were tasked with curating their own sounds in preparation for an imaginary improvised concert.Upon completion of the tasks, a semi-structured interview was conducted. The topics discussed over a duration of 20-25 minutes included: clarifications (3-5 mins), reflections (3-5 mins), participation survey follow-up (3-5 mins), comparison of personal practices to the activity (3-5 mins), and any additional questions (3-5 mins).

4.3 Data Capture and Interpretation

The data collected in this study comprised the pre-activity survey, audio and video recordings of the activity, logs of sculptures and sculpts during the activity, logs of percussion block state changes throughout the activity, post-activity discussion audio recordings, and supplementary photo/video documentation of sculpts as required. Activity and interview transcripts were annotated by Armitage following thematic analysis [2]. A tagging schema was employed, based on an array of research questions related to the PhD thesis topic, and these were complemented with themes that surfaced through iterative thematic analysis. This paper delineates a subset of those emergent themes associated with specific sculpting techniques, encompassing trial and error, timbral limit-finding (exploring the boundaries of the sculpting response), systematic approaches to sculpting, and intuition as a guiding factor.

5. OUTCOMES

In this section, we present a selection of findings from the study, which are fully reported in [2]. Our focus here is on the outcomes related to the ordered patterning of sculptures. Although the apparatus did not offer any means to visually recall sculpts, participants were able to create sequences of sculpts exhibiting distinct patterns. We identified these patterns post hoc by generating matrices of video stills for each session.

The sculpting motifs observed in these episodes served as pattern components, which were repeated and varied spatially across the sculpting surface. Over time, these patterns were iterated, occasionally with remarkable order and consistency, spanning a relatively large number of sculpts. Unfolding inconspicuously, the patterns appeared to adhere to implicit rules, and even in cases where they were less visually prominent, logical connections between sculpts could be discerned.

5.1 Sculpting Examples

Figure 3 demonstrates various sculpting patterns observed in the study. The sculpting sequence from **L4** (top right) demonstrates continuous morphing between different pattern motifs, and overlapping/combination of multiple patterns simultaneously. The top right shows three examples of patterned sculpting; **L1** took progressive slices away from a vertical cylinder, **L33** sequenced the combinations of a binary pair, and **L4** moved a line across the length of the surface.

The bottom left example shows abridged sculpting patterns from **L16**. In the top row, a group of clay bridges of different lengths are laid out parallel to each other, in different combinations, and sorted in different ways. In the

Figure 3: Top left: sculpting sequence from **L4**. Top right: three examples of patterned sculpting from **L1**, **L33**, and **L4**. Bottom left: sculpting patterns from **L16**. Bottom right: sculpting patterns from **L32**, **L20** and **L25**.

second row, small roundish pieces are arranged with increasing density before being replaced with bridges which have been flattened into lines and distributed in parallel again. In the third row, a large, surface-covering piece of clay has been cut into thin bands, some of which are then removed, and subsequently the negative space between them is varied.

The bottom right examples show sculpting patterns from **L32**, **L20** and **L25**. **L32**, in a sequence that was representative of a number of others, took small similar sized, roundish pieces and sequentially added them to different corners and sides. **L20** worked with two small pieces, moving them together to trial different symmetries and reflections. **L25** varied the distances of four pieces between the corners and the centre, maintaning a radial symmetry.

5.2 Systematic Making Methods

Participants commented on ordered or systematic approaches, either in reference to what they actually did during the session, or what they would do if they had more time:

> **L7**: “In my experience with handmade electronics, sometimes you already prepare three different values of things so that you can just switch, plug and play, and just see the difference. For example, values of resistors [...] When you are comparing things, and results of materials, you have to create something that is variable and something that’s constant. So for example the triangle [of clay] would be the constant, but then vary the shape or the feeling, like an empty one or a solid one.”

> **L13**: “I think if I would be able to sit down and actually have a method of doing it, I would probably start with the number of items. And then the area, basically test really scientifically on each parameter, like I have a compare group of having one constant variable and then compare all of them.”

> **L18**: “When I put three of them [pieces of clay], it was kind of like so I could see it. But if I had put like six or more, and then it would have been hard to like detect it. But with three it’s kind of keeping it simple, so I can hear a clear difference, and I don’t have too many combinations of all the on/off.”

While **L25** did not exhibit distinctive patterns visually in their sculpting, they relied heavily on their physics background and scientific mindset to develop organising principles for their process, and readily described how they felt their ability to test their theories and models was improving:

> **L25**: “I felt like even though I basically had started to assemble a still somewhat misty model of what I could be playing with and the parameters and how it would affect it - and I don’t feel like I got any better, and objectively, I probably didn’t - I at least had a model, which maybe after five more tests I would have totally thrown away. I still felt like I was taking baby steps, and just being like, no that theory doesn’t matter. You’ve got such few data points that there’s so many different curves that can fit those data points in terms of like finding an underlying theory for how to affect the sound.
>
> I mean, I felt sort of more, more comfortable in that process. I felt more confident, because I felt like even though I didn’t have much faith in my model, I felt like I was getting to a slightly more advanced stage of being able to test the model. There comes a point where actually you just have to accept floundering is probably better.”

6. DISCUSSION

6.1 Geometric Algorithmic Pattern and Visuospatial Cognition

There were no prompts specifically related to geometrically patterned sculpting, and participants who engaged in it did not provide detailed remarks, either during the sculpting process or upon reflection afterwards. Despite this, numerous instances of geometric patterning occurred. Considering the patterns' invisibility during the session, apart from in working memory, this implies that participants were not always consciously aware of the patterns they were creating.

Nonetheless, various reasons for sculpting geometrically can be contemplated, whether arising subconsciously or consciously, all of which seem to be connected to an intention to comprehend the sculpting system. This approach might have been a response to seemingly random or chaotic outcomes, enabling participants to maintain a systematic or consistent element in their sculpting. It might have facilitated the physical reversal of a sculpt and the return to a previous state, as well as simplifying the process of recalling sequences of sculpts. It could have allowed for the comparison of input patterns with potential output patterns, suggesting predictable control strategies and fundamental principles. The patterns might have increased the efficiency of experimenting with specific sculpting ideas and testing assumptions in logical combinations, thereby enabling procedural validation or elimination of these ideas. These patterns may have permitted the creation of intricate results from simple, memorable rules, facilitated the exploration of a motif's implications, and fostered the inventive generation of new sculpting motifs and approaches in an aesthetically appealing manner. Lastly, participants might have employed patterns in this manner due to their similarity to approaches adopted in their own practices.

These geometric patterns' existence implies that certain other processes must have been present in some form. In particular, it seems reasonable to assume that an element of design planning occurred, even if it was implicit, rapid, fragmentary, and ad-hoc. Planning sculpts could involve envisioning patterns as extrapolations of existing motifs or other background references, concretising assumptions into combinatorial spatial logic and temporal procedures, and deciding on suitable levels of granularity and frequency of change relative to time constraints. While it is not possible in hindsight to judge the extent to which any of these reasons were influential, they indicate, to varying degrees, an approach of projecting order onto the sculpting process via geometric patterns. The ease and fluidity of engaging with the system in this way suggest that it was a highly intuitive approach.

6.2 Algorithmic Informality and Non-Notated Patterns

There existed an improvisatory element in patterned sculpting, in which geometric patterns underwent permutations based on implicit rules and procedures. These rules and procedures were simultaneously being modified, composed, overlapped, and morphed as they were being adhered to. This ongoing engagement with the material manifestations of patterns and procedures bears resemblance to artistic live coding, where algorithms are rewritten during their execution [3].

Such alterations may have resulted from the participant reacting to an aspect of the environment, such as the sounds produced, the condition of the clay, or the passage of time (with regular reminders provided by the researcher). Equally, these changes could have been prompted by the participant's internal reflection on the patterns they were following. Clearly, a dialogue of considerable sophistication and fascination was taking place between the participants and the sculpting system.

Some of these patterns and temporal sequences exhibited an unmistakably algorithmic character, with such precision that their formal notation is not beyond the realm of possibility. Moreover, the repeated use and variation of clay motifs suggest a certain level of codification within the sculpting vocabulary, which would be difficult to characterise as entirely non-symbolic or non-representational. Collectively, these outcomes appear to align with Boria's definition of a notation system, as reviewed earlier [7]. However, since the participants did not explicitly notate or externally represent their sculpting procedures apart from their direct manifestation as clay sculptures, we cannot consider these observed algorithmic behaviours as fully formalised. In this instance, the procedures were informally implied by the relationships between sequential configurations of clay and only became truly visible afterwards through the arrangement of video frame sequences. Crucially, any such "program" governing the sculpting process remained tacit. The extent of real-time abstract reasoning and formalisation, as well as the participants' self-awareness of it, cannot be fully determined from this study design, aside from the post-activity interview data. So, were these sound sculptors engaging in algorithmic patterning or not? If not, did their process transform into algorithmic patterning when we, as researchers, interpreted and pseudo-notated it?

6.3 Proto-Algorithmic Pattern in Live Coding Research

Due to the compelling nature of the observations made, we posit that it is valuable to assume the presence of a tacit and informal representation of algorithmic patterning in this case. We define this as *proto-algorithmic patterning*, referring to any activity in which material engagement with symmetry results in the emergence of symbolic and procedural motifs that initiate dialogue between the variant and invariant qualities of experience. As it lacks the formalism required of an algorithm, we also contrast proto-algorithmic patterning with simple heuristics.

Heuristics can be both formal (e.g. Simon's satisficing heuristic) or informal (e.g. affect heuristic) [4], but they tend to oversimplify the continuum of formality into fast (heuristic) versus slow (non-heuristic) strategies. Furthermore, they do not account for the role of the body [11] or the generativity of materials [1] in processes related to algorithmic patterning. While it may be possible to categorise participant responses using known heuristics, doing so would ultimately abstract away the body and materiality. Instead, we aim to emphasise the necessity of concreteness and situatedness that these elements bring.

Further research is needed to explore this topic, and in this paper, we consider it from the perspectives of live coding practice, systems design, and research. This work substantiates the necessity for spaces such as the aforementioned HLCI, where participants have been venturing "beyond the screen" and delving into their own tacit dimensions, most recently examining bodies, voices, and pedagogies. By emphasising embodiment and materiality as underexplored areas in live coding [5], the concept of proto-algorithmic patterns may potentially assist practitioners in reclaiming some of the power typically attributed to computing machines as their own. Furthermore, we encourage live coding practitioners to contemplate informality and visuospatial imagery in their practice, and to share more of their internal states for their own benefit, as well as to inspire system design and research.

In the design of live coding systems, the pursuit of deeper exploration into informal and geometric algorithmic patterning necessitates the development of systems that can support such endeavours. These systems ought to facilitate underdetermined, ambiguous, and sketch-like programs, prioritising flexibility over rigidity. Large language models (LLMs) can now generate highly plausible interpolations of training data, allowing users to turn vague natural language into code [18]. However, these systems are still far from supporting the kind of freeform, exploratory, and generative sculpting that we observed in this study. Rather than manifesting as closed environments, we would like to see new systems that actively participate in everyday reality, as exemplified by the Dynamicland project[5].

The "languages" employed in these systems need not be limited to traditional written, textual, or spoken forms; they can also be expressed through drawing, sculpting, and visual demonstrations. Potential starting points for these languages include geometry [19], geometric deep learning [20], differentiable programming [21], or self-organising systems [**?**]. Furthermore, these systems can draw inspiration from concepts such as textual inversion to bridge the gap between the brain's linguistic and visuospatial capabilities [22].

If live coding practice and systems were to incorporate some of these ideas, the scope of live coding research would broaden, potentially attracting more external experts from fields such as psychology, cognitive neuroscience, mathematics, and beyond. Instead of focusing solely on fully formalised algorithmic patterning, researchers could investigate algorithmic patterning as a spectrum of formality. Approaches to a more detailed and controlled examination of the emergence of algorithmic patterns and their relationship to embodied experience could, for instance, build upon this work while establishing more rigorous apparatus, protocols, and methods.

6.4 Proto-algorithmic pattern in DMI design research

It was a surprising discovery to us that a concept from musical live coding could aid in decomposing DMI design activity. However, live coding is a term still being used by non-arts based communities.[6] Live coding and the quick feedback and iteration loops it provides are well understood in these fields as being essential for software design and craft, yet over time these fields have become separated and distant from live coding in the arts. As we have argued elsewhere in assessing the liveness of DMI design tools [14], digital luthiers need Tanimoto's liveness levels [23] too for the sake of their craft. In this paper we have shown that proto-algorithmic pattern is perhaps an essential and overlooked process in DMI design, in the way that design can be conceived of as a search problem, and that live coding tools could be repurposed as DMI design aids.

7. CONCLUSION

In this paper, we combined our experiences as live coders and DMI design researchers to investigate a mode of informal, geometric algorithmic patterning that operates prior to the formal manipulation of symbols in notating an algorithmic pattern. We refer to this as *proto-algorithmic patterning*, alluding to the pre or proto-symbolic mode of thinking in tacit and embodied exploratory situations. A significant aspect of musical performance involves motor-memory, intuitive responses to novel situations, and tacit knowledge of the tradition, instrument, fellow musicians, and the unique musical habitus in which the performance occurs.

Writing an algorithmic pattern in an algebraic notational language, such as those used in most live coding environments, constitutes a linguistic mental activity, somewhat removed from visuospatial experience, the body, and its motor memory. There exists a level of algorithmic thinking that occurs on the cusp of when we begin to operate with symbols in a step-by-step instructional manner, and this mode of thinking is equally relevant in creating algorithmic patterns. Proto-algorithmic patterning is an activity of exploration and discovery of geometric ideas and, in this case, material-to-sound relationships.

What are the implications of introducing proto-algorithmic patterning as a consideration in the work of designing algorithmic interfaces for artistic expression? We tentatively hypothesise and encourage the reader to consider that the next SuperCollider[7] or TidalCycles[8] might not be an algebraic, linguistic, primarily discrete, and fully formal programming system. Instead, it could incorporate non-human agents, plants, animals, analogue computing, artificial life, and differentiable or probabilistic programming as part of an algorithmic pattern system, thus opening up towards embodied reasoning and embracing continuous lived experience. Learning with and understanding proto-algorithmic

---

[5] `https://dynamicland.org`

[6] See `https://liveprog.org/` and `https://programming-conference.org/`

[7] `https://supercollider.github.io/`

[8] `https://tidalcycles.org`

patterning would then hopefully emerge through bodily movements and physical trial-and-error, rather than learning through language via pre-existing, discrete symbols. The point at which proto-algorithmic patterning converges with algorithmic thinking through formal notation is not necessarily a distinct dividing line; rather, we encourage the conception of a continuum from informal to formal. Lastly, we emphasise that geometric algorithms and visuospatial cognition are the primary materials with which we believe proto-algorithmic patterning can be further investigated.

Acknowledgments

Thanks for feedback on drafts of this paper to Alex McLean, Isaac Cohen and Victor Shepardson. Thanks to the anonymous reviewers for their comments and suggestions.

This research is supported by the European Research Council (ERC) as part of the Intelligent Instruments project (INTENT), under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101001848).

This research was also supported by EPSRC in the UK, under the grants EP/L01632X/1 (Centre for Doctoral Training in Media and Arts Technology) and EP/N005112/1 (Design for Virtuosity).

8. REFERENCES

[1] A. Mclean, "Algorithmic pattern," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, R. Michon and F. Schroeder, Eds. Birmingham, UK: Birmingham City University, Jul. 2020, pp. 265–270.

[2] J. Armitage, "Subtlety and detail in digital musical instrument design," Thesis, Queen Mary University of London, Apr. 2022.

[3] C. Nilson, "Live Coding Practice," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, New York City, NY, United States, 2007, pp. 112–117.

[4] T. Gilovich, D. Griffin, and D. Kahneman, *Heuristics and Biases: The Psychology of Intuitive Judgment*. Cambridge University Press.

[5] S. Salazar and J. Armitage, "Re-engaging the Body and Gesture in Musical Live Coding," in *Proc. New Interfaces for Musical Expression*, Blacksburg, Virginia, USA, 2018.

[6] W. Mitchell and M. Hansen, *Critical Terms for Media Studies*, ser. Critical Terms. University of Chicago Press, 2010. [Online]. Available: https://books.google.is/books?id=eb4HDw0CklEC

[7] A. F. Blackwell, E. Cocker, A. McLean, and T. Magnusson, *Live Coding: A User's Manual*. MIT Press, 2022.

[8] A. McLean, "Artist-programmers and programming languages for the arts," Ph.D. dissertation, Goldsmiths University of London, London, United Kingdom, 2011.

[9] J. Rohrhuber, "Algorithmic Complementarity, or the Impossibility of "Live" Coding," *Collaboration and learning through live coding*, vol. 3, no. 9, pp. 130–168, 2013.

[10] W. W. Gaver, J. Beaver, and S. Benford, "Ambiguity as a resource for design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2003, pp. 233–240.

[11] A. Clark, *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*. Oxford University Press, 2015.

[12] B. Victor, "Stop drawing dead fish," in *ACM SIGGRAPH*, San Francisco, CA, USA, 2012.

[13] R. Eglash, *African Fractals: Modern Computing and Indigenous Design*. Rutgers University Press, 1999.

[14] J. Armitage and A. McPherson, "Bricolage in a hybrid digital lutherie context: A workshop study," in *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*. Nottingham United Kingdom: ACM, Sep. 2019, pp. 82–89.

[15] J. Armitage, T. Magnusson, and A. McPherson, "Design Process in Visual Programming: Methods for Visual and Temporal Analysis," in *Proc. Sound and Music Computing Conference*.

[16] ——, "A Scale-Based Ontology of Digital Musical Instrument Design," in *Proc. New Interfaces for Musical Expression*.

[17] ——, "Studying Subtle and Detailed Digital Lutherie: Motivational Contexts and Technical Needs," in *Proc. New Interfaces for Musical Expression*.

[18] S. Barke, M. B. James, and N. Polikarpova, "Grounded Copilot: How Programmers Interact with Code-Generating Models," vol. 7, pp. 78:85–78:111. [Online]. Available: https://dl.acm.org/doi/10.1145/3586030

[19] S. Kaliski, I. Reese, and M. Goethals, "Crosscut: Drawing Dynamic Models," https://www.inkandswitch.com/crosscut/, Mar. 2022.

[20] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. [Online]. Available: http://arxiv.org/abs/2104.13478

[21] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," *arXiv:2001.04643 [cs, eess, stat]*, Jan. 2020.

[22] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion," Aug. 2022.

[23] S. L. Tanimoto, "A perspective on the evolution of live programming," in *International Workshop on Live Programming*, 2013.