



Anatomical Intelligence: Live coding as performative dissection

JOANA CHICAU¹  and JONATHAN REUS² 

¹Creative Computing Institute, University of Arts London, London, UK

Email: j.chicau@arts.ac.uk

²Sussex Humanities Lab, University of Sussex, Brighton, UK

Email: j.reus@sussex.ac.uk

This article describes the method of ‘dissective’ live coding, as developed through the artistic-research project Anatomies of Intelligence. In this work we investigate how live coding can be used as an approach for performative explorations of a data corpus and a machine learning algorithm operating on this corpus. The artistic framework of this project collides early Enlightenment-era anatomical epistemologies with contemporary machine learning, creating a fertile space for novel, embodied artistic methods to emerge. We engage audiences in an immersive, live-coded experience where image and sound are driven by our dissective approach, revealing the underlying rhythms and structures of a machine learning algorithm running live on an artist-made dataset. To support these performances we have developed a custom browser-based software, the Networked Theatre, used for both hybrid in-person/online audiovisual performances. In this article we describe this work and reflect on our experience as performers and audience feedback, which suggests that our dissective method of live coding, based on examining ‘ready-made’ algorithms, offers a unique experiential entryway into the bodies of machine learning and data corpi.

1. INTRODUCTION: ANATOMIES OF INTELLIGENCE

Anatome and Anatomia: the words signify as much as a dissection. But being taken for an art and applied to a certain object, they signify an artificial dissection of that object in such manner as may most conduce to the perfect knowledge of the same and all its parts. (Cunningham 1993: 15)

Transparency is a concept often addressed in discussions around live coding. The 2004 TopLap manifesto stated: ‘Obscurantism is dangerous. Show us your screens’ (Ward, Rohrhuber, Olofsson, McLean, Griffiths, Collins and Alexander 2004: 290). By showing our screens, we run counter to the acoustic wall of the musician-with-laptop paradigm that dominated electronic music concerts at the time the manifesto was written. However, even visible code includes many levels of abstraction, from functions and variables to patterns and classes. This implies that there are always some decisions that are hidden; that we performers in part choose when and how to

reveal parts of our thinking. On a practical level, we offload rote and repetitive labour to abstractions that make live coding ‘faster and less of an inventing-the-wheel-in-front-of-a-live-audience process’ (Blackwell, Cocker, Cox, McLean and Magnusson 2022: 243). On an artistic level, the process of ‘making visible’ one’s reasoning is not always about transparency, but can be dramaturgical, and sometimes be used (wilfully) to create further opacity, desirable confusion and bewilderment (Cocker 2016: 109). Transparency in live coding is not a simple binary of visible and hidden (screens), and is a complex amalgam of creative and technical decisions that constitute the possibility space of a performance.

In this article we describe Anatomies of Intelligence (AoI)¹ – an artistic research project whose fundamental approach lies in operationalising such a possibility space, while also referencing another practice of selectively revealing parts of complex wholes – the anatomical and surgical theatres of eighteenth-century Europe. The European ‘Era of Enlightenment’ was a time when Dutch anatomists such as Frederik Ruysch, Jan Swammerdam and Reinier de Graaf invented techniques to dissect and transform human organs into ‘preparations’ that could be preserved and studied for generations to come (Knoeff and Zwijnenberg 2015). To better understand the collections of anatomical specimens from this time, the historian Marieke Hendriksen has developed the epistemological concept of *aesthesis* to describe ‘the faculty or power of sensation’ of a given specimen (Hendriksen 2015: 10). According to Hendriksen, *aesthesis* helps to explain the value decisions made by those who created these collections. Notably, *aesthesis* emphasises that knowledge is gained by being hands-on with, and in proximity to, anatomical specimens.

No doubt, the Enlightenment was also a time of great change in how European intelligentsia understood truth and knowledge. We are now experiencing a similar shift, the ‘epistemological sea change of the

¹Chicau, J. and Reus, J. 2018. Anatomies of Intelligence. <https://anatomiesofintelligence.github.io/> (accessed 29 May 2023).

21st century, driven by massive-scale data collection, pattern fitting and prediction algorithms' (Reus 2021: 100). AoI's conceptual anchor point is the drawing of connections between these two eras, and in doing so considering how *aesthesis* might be used to understand complex computational objects such as algorithms and datasets. We use live coding as a gateway to this *aesthesis*, allowing the possibility of being 'hands-on with' and 'in proximity to' these digital bodies. We use sound and visuals to give shape and presence to them, and to reveal their forms not all at once, but selectively through a process of on-the-fly composition and theatrical worldbuilding.

AoI is a broad artistic project with many outputs, including workshops, performances and installations (Cox and Soon 2021: 235–6). In this article we focus on the live coding performances where 'dissection' takes centre stage as method and metaphor. In the sections that follow we attempt to create a context of other artists and researchers working in related domains of embodiment and data-driven art, we describe our live coding toolkit,² our dissective approach and the kinds of visual, sonic aesthetics and performance styles that emerge from our method.

2. EMBODIMENT, MACHINE LEARNING AND LIVE CODING

There is growing interest within live coding research communities on bringing machine learning into live coding tool sets. Examples include FluCoMa (Tremblay et al. 2022) and the MIMIC project,³ which have been shared across various communities of artists, technologists and educators for exploring machine learning in their creative work. Another example is Sema, a web-based playground for integrating JavaScript-based machine learning libraries with custom live coding languages (Bernardo, Kiefer and Magnusson 2020). Many artists are also building their own individualised tools and processes, such as in *The Machine is Learning*,⁴ by artist Marije Baalman, who performs a theatrical piece whereby a machine is trained to detect simple gestures yet repeatedly fails to do so accurately. The choreographer and live coder Kate Sicchio has also been working with machine learning algorithms. In her piece *Untitled Algorithmic Dance*,⁵

she feeds images of bodies in motion to a t-Distributed Stochastic Neighbour Embedding (t-SNE) algorithm, producing new choreographic scores in live performance. Meanwhile, recent artistic research initiatives such as *Algorithms that Matter* have brought together multiple artists to investigate the ways in which unique engagements with algorithms, including machine learning, structure sonic expression (Pirò and Rutz 2022).

Authors in the field of human–computer interaction have for many years been reflecting on forms of embodiment relating to computing, which has only intensified in recent years as data-driven approaches have become widespread. In the words of Paul Dourish, 'embodiment is the common way in which we encounter physical and social reality in the everyday world' (Dourish 2001: 100). Within data-driven machine learning, Catherine D'Ignazio and Lauren Klein argue for embracing emotion and embodiment as part of the principles of Data Feminism. D'Ignazio and Klein emphasise the value of embodied modalities to making data-science legible, highlighting how 'activating emotion, leveraging embodiment, and creating novel presentation forms help people grasp and learn more from data-driven arguments, as well as remember them more fully' (D'Ignazio and Klein 2020: 88). This feminist perspective is one that Jonathan Reus has also discussed in relationship to *AoI* and other artworks, writing that '[in data-driven art] there must be a left hand of context and care to steady the right hand of the impulse to collect and discriminate' (Reus 2021: 100).

More generally, Reus's series of works beginning in 2012 under the title *iMac Music* address the tension between the human body and the material strata of computation. In these performances Reus deconstructs running obsolete computers, using sound amplifying probes to bring out the rhythmic and timbral electrical signatures of everyday software processes. This approach physically implicates the performer at multiple points in the technical stack, blurring any neat dichotomies between hardware and software and even causing *iMac Music* to be a somewhat controversial entry into the canon of live coding (Baalman 2015; Han and Reus 2018). However, as cultural theorist Sally Jane Norman notes, 'live coding demands specific kinds of engagement from its audiences. More than actual coding literacy . . . it demands willingness to try and sense the dramatic competition between autonomously evolving algorithms and human interventions' (Norman 2016: 4). For her, this relationship between human interventions and algorithms is ontological, and a performance like *iMac Music* creates a 'theatre of machine anatomy', whose relationship between performers, audience and algorithms could be seen as one spiritual precursor to the dissective approach of *AoI*.

Specific terms and tactics have been emerging for making sense of modern machine learning and

²Chicau, J. and Reus, J. 2018. Anatomies of Intelligence Online Repository. <https://github.com/anatomiesofintelligence> (accessed 29 May 2023).

³Grierson, M., Yee-King, M., Fiebrink, R., Magnusson, T., Collins, N., Kiefer, C., et al. 2019. <https://mimicproject.com/about> (accessed 12 September 2022).

⁴Baalman, M. 2019. *The Machine is Learning*. <https://marijebaalman.eu/projects/the-machine-is-learning.html> (accessed 29 May 2023).

⁵Sicchio, K. 2017. www.sicchio.com/work-1/untitled-algorithmic-dance (accessed 29 May 2023).

autonomous systems with a focus on embodiment. *Graspable AI* (Ghajargar, Bardzell, Smith-Renner, Höök and Krogh 2022) proposes the use of physical artefacts and material manifestations as a relational way of understanding and interpreting algorithmic systems; while the related field of *Experiential AI* (Hemment, Aylett, Belle, Murray-Rust and Luger 2019) proposes to make algorithmic mechanisms understandable through felt experiences. Experiential AI addresses the challenge of ‘finding novel ways of opening up the field of artificial intelligence (AI) to greater transparency and collaboration between human and machine’, and in doing so aims to ‘dispel the mystery of algorithms and make their mechanisms vividly apparent’ (ibid.: 25). We interpret Experiential AI as a direct call to artists to explore novel ways for making AI systems decipherable.

As another example following an embodied approach, the artist Memo Akten has been incorporating slow, meditative ‘spiritual journeys’ within his live performances using deep artificial neural networks, such as in the work *Deep Meditations: Morphosis*.⁶ In an article from 2015, Akten refers to architect Liam Young’s concept of *Data Dramatization*, which aims not only to present a dataset in a legible way, but also to provoke an emotive or empathetic reaction. In this sense, dramatising data entails both analysing and extracting stories that are in the data (Akten 2015). In live coding, Nick Collins has introduced a related notion of ‘dramatising a computer algorithm’ (Collins 2011: 207) and live coding as a form of ‘perturbation’. He gives the example: ‘let people place themselves into order of height, using an agreed-upon sorting algorithm; a live-coding twist would be to perturb the algorithm halfway through, perhaps when one participant wanders off to fetch a drink’ (ibid.: 207). While humans physically acting out an algorithm like this is a very literal form of dramatisation, one might extend this concept to consider that any algorithmic process operating on some kind of data, when made performative and sensible, is a form of *dramatisation*, whereby the algorithmic operations and dataset become the driving dramaturgical element of an artistic work. Other notable examples are the sonified sorting algorithms of the Institute for Algorhythmic⁷ or the visual pixel sorting algorithms popularised by Kim Asendorf during the early years of Glitch Art (Penney 2016).

As mentioned earlier, we see diverse approaches from artists and designers in relation to embodiment and dramatisation of algorithms and data. And in

the field of live coding research, we identify the development of tools and ecosystems for integrating machine learning to produce music, visuals and choreography. However, despite the work of artists such as Baalman, Reus and Sicchio, and the Algorave scene’s focus on danceability, our impression is that live coding remains with one foot in the shadow of a Cartesian mind–body dualism. The live coding literature itself is full of such implicit bias, with references to ‘making thought visible’ and ‘thinking in public’ (Blackwell et al. 2022: 6). This is, perhaps, an unavoidable side-effect of the symbolic languages that make up programming languages at large. Another potential source of this duality could be traced to European traditions of art music, with their hierarchical separation of composer (mind) and instrumentalists (bodies). This historic inertia has been part of computer music from its inception and, even within the relatively open and non-hierarchical artistic world of live coding, may find subtle ways to linger. But while the sources of this sneaky mind–body dualism are likely complex and multifaceted, it should be noted that there are many aspects of programming that are not functionally mandated and could easily be changed to open new pathways of experience. These include, for example, the vocabulary and metaphors used within programming keywords to describe behaviours and functions. One of the most explicit examples of such a functionally irrelevant naming convention is found in the HTML standard, which uses the tags `<head>` and `<body>` to separate a website’s invisible metadata (head) from its visible content (body).

Similarly, the field of AI suffers from the rampant use of implicitly dualist vocabulary, loaded as it is with brain and human cognitive terminologies. Terms such as ‘neural networks’, ‘machine learning’ and even ‘artificial intelligence’ do the dual disservice of both anthropomorphising computational systems and doing so in a way that creates a distinctly dualist conception of computation. As George Lakoff and Mark Johnson explain in their seminal experiential design text *Metaphors We Live By*, our conceptual systems and metaphors ‘structure what we perceive, how we get around in the world, and how we relate to other people’ and play a ‘central role in defining our everyday realities’ (Lakoff and Johnson 2008: 12). We would add that metaphors also structure how technological realities are constructed. We therefore explicitly act against this brand of AI-anthropomorphism by avoiding the classic metaphors relating to human brains and cognition, opting instead for a novel set of metaphors drawing on anatomical structures, physiological systems and the gestures involved in their examination.

We strive to include bodily awareness as part of *AI*, defining ‘bodily awareness’ as ‘sensemaking through

⁶Akten, M. 2019. www.memo.tv/works/deep-meditations-morphosis/ (accessed 12 September 2022).

⁷Miyazaki, S. 2008. https://monoskop.org/Institute_for_Algorhythmic (accessed 12 September 2022).

one's own body movements, sensory attention and a reflective awareness of the position and composition of one's own body' (Tapparo and Zappi 2022). We do this through complex performances that bring together sonic, visual and voice-guided experiences centred around intimacy and proximity (Chatzichristodoulou and Zerihan 2012) that aim to create feelings of being close-to objects under examination as well as other audience members, and, in the words of Josephine Machon, 'enable practitioners and audience members alike to tap into pre-linguistic communication processes and engages with an awareness of "the primordial"' (Machon 2009: 1).

In the next section we describe the staging of an AoI performance, and describe the browser-based software we have developed to facilitate the use of live-coding in these performances.

3. AN OPERATIONAL STAGE: THE NETWORKED THEATRE

AoI takes place in a hybrid audiovisual setting, 'hybrid' meaning the superposition of an in-person performance alongside a globally accessible browser-based performance. The link between these settings is a web platform we call the Networked Theatre, which is accessible to online audiences via a URL and/or projected in the venue where a performance takes place. The Networked Theatre is a web application built using HTML/CSS/JavaScript, within which audio and visuals are triggered on-the-fly by performers by remotely sending JavaScript code from their own computers to audience's browsers. The web application includes custom JavaScript libraries that makes this remote browser coding possible, as well as libraries for accessing the AoI dataset and for running real-time machine learning algorithms. The AoI dataset,⁸ as of this writing, includes 99 entries, which are a mixture of images and texts collected from AI research publications and anatomical research archives, including arXiv, the Wellcome Collection, Wikimedia Commons, towardsdatascience.com and the Leiden Anatomical Collections.

The scenography of *AoI* is inspired by the anatomical theatres of the eighteenth century; large circular amphitheatres whereby an anatomist would perform dissections of human and animal cadavers upon a central platform, surrounded by onlookers (Knoeff and Zwijnenberg 2015) (Figure 1). In the browser, spherical grids are a repeating visual motif of the performance, and are juxtaposed against 2D overlays of code snippets transmitted remotely from the performers' code editors to the browser (Figure 2).

Voice dialogues from the performers are spatially processed on-the-fly using WebAudio nodes⁹ in the browser. In the physical staging, *AoI* uses projections on hanging semi-transparent gauze screens that surround the performers and audience, and a central circular floor projection – a homage to the dissection platform of the anatomical theatre (Figure 3). The surrounding gauze projections suggest an amphitheatre-like architecture while also hinting at the gauze-like fabrics used decoratively in eighteenth-century anatomical preparations.

The staging creates an audience–performer relationship where the audience are not treated as passive viewers, but rather as participants who actively take part in the dissective exploration of the phenomena under study. Online, the audience visiting the Networked Theatre are able to explore the dataset entries with their mouse as with any webpage, clicking on entries and moving the perspective view around as the stage and display elements are constructed and modified by the performers. There is also a chat function, where audience members can send messages to one another and to the performers. In situ, the audience lays on custom-made inflatable sculptures created by artist and researcher Dominique Savitri Bonarjee. In this scenario the performers act as facilitators, guiding the audience and narrating the process of inspecting the dataset and algorithm. At certain moments of the performance, the audience are invited to gather around the central circular projection (Figure 3). This performative framing of audience–performers–algorithms is similar to the way in which more intimate historical anatomical demonstrations blurred the boundaries between experts and students.

In her solo live coding performance practice, Joana Chicau manipulates the web browser, calling and manipulating JavaScript functions from a glossary of code blocks inspired by choreographic notations (Chicau 2017). In these performances, she uses the developer console of the web browser as a real-time performance interface, using it to directly alter the underlying code and visual appearance of web pages. Chicau effectively re-appropriates the developer console and transforms the web browser itself into a live coding instrument; an approach that became fundamental to the conceptualisation of the Networked Theatre. With this platform we wanted to be able to perform as if remotely accessing the audience's developer consoles, reconfiguring their browsers on-the-fly. *AoI* thus owes much to the legacy of browser-manipulation in Net Art. In our performances we often make use of browser-specific

⁸Chicau, J. and Reus, J. 2018. Anatomies of Intelligence Dataset. <https://github.com/anatomiesofintelligence/dataset> (accessed 27 May 2023).

⁹WebAudio is an open and widely supported browser-based audio standard, providing a virtual audio patching system. WebAudio provides modular 'nodes' for audio synthesis as well as DSP, which can be recombined and reconfigured in real-time.

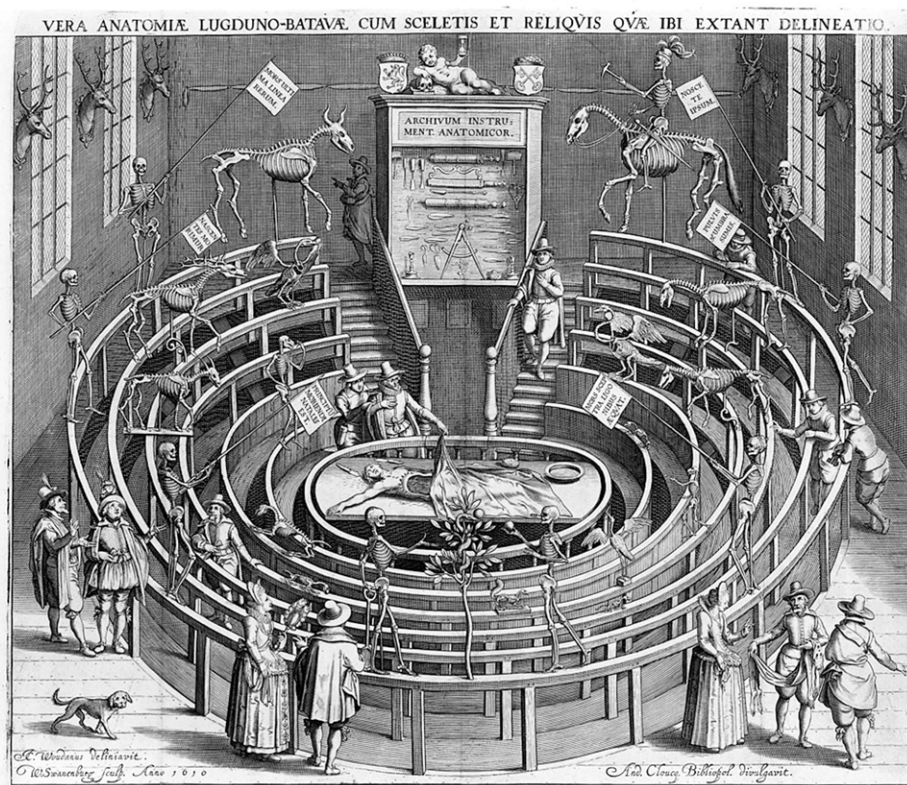


Figure 1. Leiden Anatomical Theatre, Willem Swanenburg after Johannes Woudanus, the Leiden, Anatomy Theatre (1610). Source: Wikimedia Commons, under Public Domain.

affordances such as pop-up windows that open and close suddenly, website elements that move uncannily around the screen, and the performers' enabling and disabling of visitor mouse movement. The networked dimension of the performance is also emphasised in the display of number of spectators in the top right corner of the interface, making online audiences aware of 'others' attending with them.

In the spirit of making executable code a 'meaningful formal element of performance' (Blackwell et al. 2022), the Networked Theatre displays the performers' code snippets as onscreen text, visible to the audience (Figure 4). The performers may also send non-executable lines of arbitrary text, which appear in the theatre as messages directed to the audience, similar to the way in which many live coders use comments as a way to communicate directly with the audience, engage them theatrically or banter with them. In *AoI* we interweave executable code snippets with direct plain-text language in order to maintain the narrative arc of a performance, relying on both the naming of functions and the display of verbal text to guide, inform and immerse the audience.

Within the theatre, we begin with a blank HTML canvas and slowly build the scene by adding layered

3D objects (using WebGL, a JavaScript API for rendering graphics) and 2D objects (using HTML/CSS), audio enters in the form of spoken word monologues and sonifications of algorithmic processes (WebAudio nodes). The sound design draws upon the aesthetic notion of 'viscerality', and is used as sonic markers to plot out the stages of a running algorithm.¹⁰ Importantly, our 'blank' HTML canvas is not the same 'blank slate' approach used in certain live coding performances, where 'performers attempt to start from a blank page' (Collins 2011: 207). Here 'blank slate' is closer to the use of this term in web development¹¹ to refer to templates that provide the most basic skeletal functionality upon which developers build their apps. The following are some snippets of performance code, demonstrating the commands for initiating a new performance scene, silently adjusting the audience's mouse behaviour, animating elements within the 3D scene and then finally sweeping the theatre clean of all visible dataset entries and randomising their positions:

¹⁰See section 5.3 for a discussion of sonic aesthetics.

¹¹See 'BlankSlate' the definitive WordPress boiler plate theme, <https://en-gb.wordpress.org/themes/blankslate/> (accessed 19 March 2023).



Figure 2. Performance at V2_Lab for Unstable Media (2022). Photo: Fenna De Jong.

```

initScene();
<<silent>>theatre.mouseMove = UI.followGaze
  Limited;
MOVE(view.position).to({x:0, y:20, z:130}, 7000,
  EASE.sineInOut);
MOVE(floor.scale).to({x:1, y: 1, z: 1.0}, 8000,
  EASE.sineInOut);
MOVE(floor.position).to({x: 0, y: 1, z: 6}, 2000,
  EASE.sineInOut);
theatre.sweepSurface();
theatre.randomize();
    
```

The appearance of HTML elements in the browser can also be modified on-the-fly. In the following example, in order to draw the attention of the audience

to specific data points, the performers change the visual size of an entry by animating its ‘scale’ property and adding a drop shadow. This is all part of the built-in functionality of a standard web browser.

```

document.querySelector("#monster_
  instruments").style.transition = "transform 1s,
  box-shadow 2s";
document.querySelector("#monster_
  instruments").style.transform= "scale(6)";
    
```

Having discussed the Networked Theatre and performance staging, in the next section we introduce our notion of dissection in live coding and what this means in practice.



Figure 3. Performers and audience members at V2_Lab for Unstable Media (2022). Photo: Fenna De Jong.

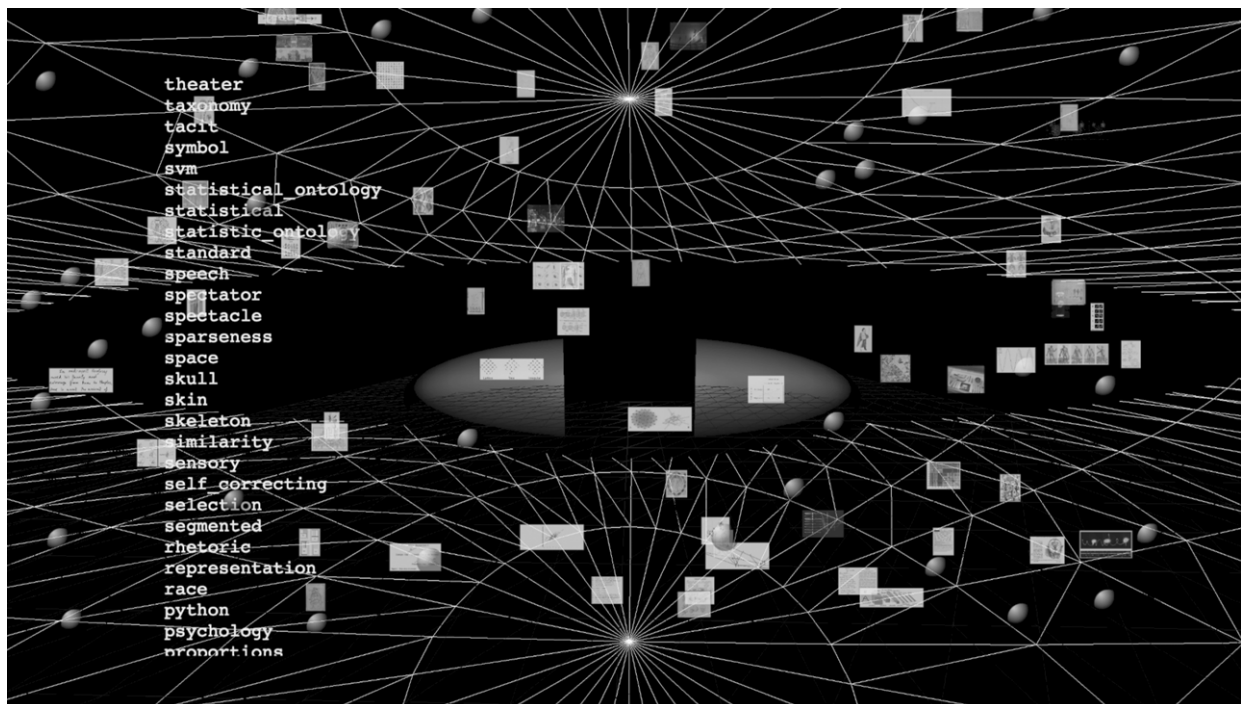


Figure 4. Screenshot of the Networked Theatre displaying the classification tags, dataset entries and spherical grids in the background.

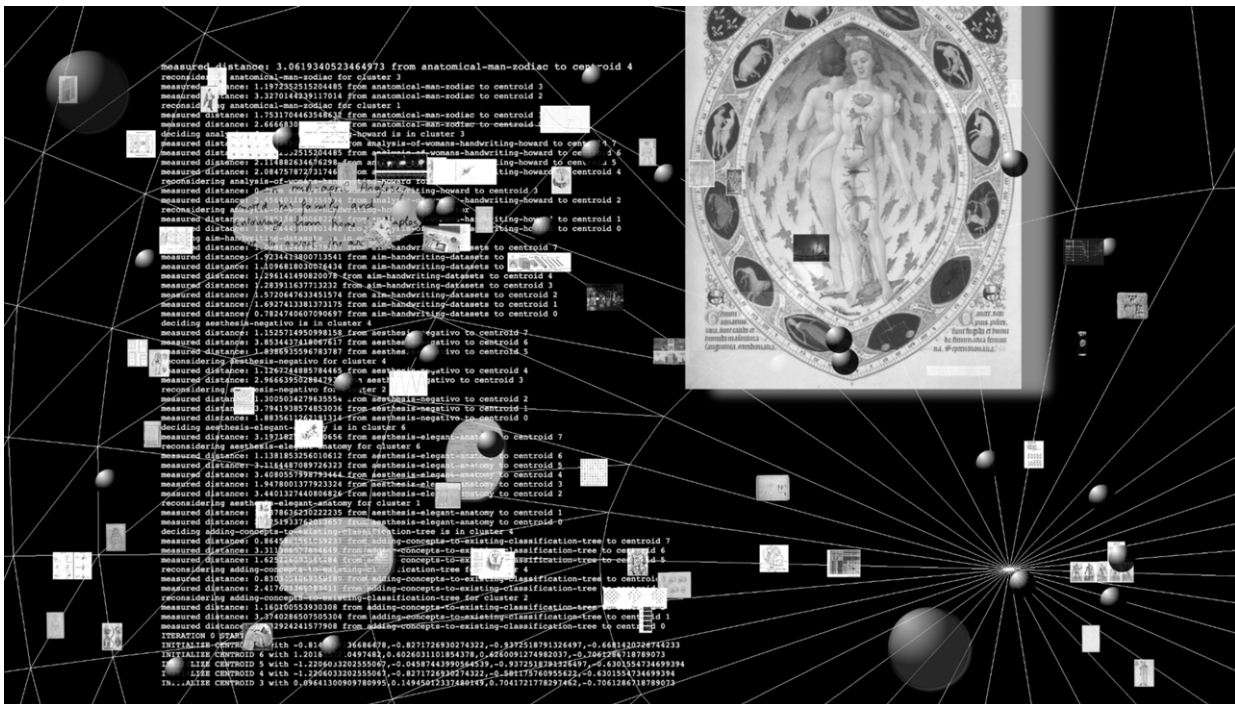


Figure 5. Screenshot of the *Networked Theatre* displaying steps of the clustering algorithm, dataset entries and spherical grids in the background.

4. LIVE CODING AS PERFORMATIVE DISSECTION

Performative Dissection is an exploratory approach to live coding that involves first selecting an algorithm and/or dataset. It is common in live coding to design ‘patterns’, algorithms that generate formal structures such as pitch sequences or rhythmic durations. In the dissective approach the structure-generating algorithm is not designed but is rather chosen from the wide selection used in research and industry. Such a ‘ready-made’ algorithm is chosen based on either dramaturgical value, or a general interest in investigating its temporal dynamics.

The algorithm we choose for examination in most *AoI* performances is the K-means clustering algorithm,¹² operating on a small custom dataset consisting of images and texts collected from historical anatomical research collections and data science publications (Figure 5). To *dissect* (dis-se-care, to cut apart) is to separate something into distinct parts for critical examination; as in the cutting of an organism (animal or plant) for the purpose of revealing, identifying and examining its organs and connective tissues. With its emphasis on examination, dissection is defined as much by how attention is directed towards experimental objects as much as it is by the act of separating the whole into categorical parts. In order for an algorithm

to be attended in such a way, it requires certain preparations to be made. Rather than using off-the-shelf libraries as is the common practice in scientific computing, we take time to ‘prepare’ the algorithm, like one might prepare a piano or an anatomical body. This preparation includes editing the code to include ‘cutting points’ where sonic and visual gestures may be inserted at any step of the algorithm, and may be influenced by whatever calculations or state is being performed at that step. As part of dissection, the performer should be able to inspect the process, zoom in, zoom out, and generally manipulate a frame of perspective – how these gestures play out in any particular dissective approach could vary, but they generally must allow the algorithm to be explored at a human-scale temporality, slowed down to the measures of beats and seconds rather than clock-ticks and nanoseconds.

The dissection of algorithmic bodies, rooted in *aesthesis*, relies on their becoming both a *felt* and *proximal* sensory reality, in being heard and seen in either physical or simulated space. Sound processes mapped to each part of the algorithm invite the audience to experience the algorithmic process with their ears and bodies, while visualisations display the dataset entries being considered and categorised at each step of the algorithm. We have found that machine learning algorithms offer a kind of data corpus suited to the dissective approach. This is due to the fact that machine learning systems often operate

¹²See section 4.1 for a discussion of this algorithm.

on very high-dimensional data – our own dataset, with its 135 features/dimensions, is a modest example. This high dimensional data can be examined from many angles, at different scales, or with different subsets of features at all steps of the algorithm. This is analogous to the immense complexity and interconnectivity of anatomical systems, and the ways in which these systems can be examined through a dissection.

Before we dive deeper into our treatment of K-means, we will discuss one last topic relevant to dissection: legibility. In an anatomical demonstration, tools such as scissors and scalpels are legible as the implements of manipulation. In live coding, legibility (Xambó 2021) of action must be created explicitly through the naming of programming abstractions such as functions and variables, or the use of various types of plain language text written for the sake of communicating with the audience. Some live coding performances, and even languages such as the music language Mercury (Hoogland 2019), put high priority on conceptual legibility by using human-readable words for parameters and functions, and by limiting the amount of code that is displayed at a given time to small snippets. *AoI* attempts to take this route by using functions and parameters named after and behaviourally resembling anatomical metaphors. This creation of a programming syntax that mixes natural language with JavaScript syntax serves a theatrical role as much as a technical one, emphasising the interplay of mechanic and human agencies, the processes and decisions of the machine learning algorithm as well as the ones from the performers (Xambó 2021).

4.1. Dissection of a K-means clustering

K-means is a cluster analysis algorithm used widely across statistics and data science. Its goal is to take a set of entities (image and text entries in our case) whose categorical distinctions are unknown and to divide this dataset into a number of groupings, or ‘clusters’. The judging criteria for an entity’s membership in a cluster is based on its similarity to other items in the dataset, each being codified as a set of numeric ‘features’. One provides the algorithm with three quantities: (1) a dataset and some means for similarity comparison, (2) the number of desired clusters K , and (3) the number of iterations of the algorithm (epochs) E to complete before stopping. K-means, like most machine learning algorithms, works iteratively, with each iteration bringing us a tiny bit closer to a mathematically optimal solution.

Despite a strong curiosity to try out other algorithms, we have found it extremely rewarding to spend an extended amount of time with K-means. Even this simple learning algorithm is capable of producing highly complex and varied temporal sequences over time. And while we are continually surprised by the complex dynamics of K-means, discovering this

complexity involved a process requiring prolonged sonic and performative experimentation. K-means has also served our audiences well – due to its simplicity, it is an accessible entry point for a lay audience to connect with the broader poetics of the work, probing ontological assumptions relating to classification and categorisation, space and distance, optimisation, dimensionality and orientation in AI and anatomy. All these concepts would be relevant to any critical artistic intervention into machine learning algorithms, yet the accessibility of K-means allows them to come more readily to the foreground of a performance.

A generalised, step-by-step outline of the K-means algorithm:

1. standardise dataset features according to mean and standard deviation of the entire dataset
2. create initial, random feature values for the K cluster centroids
3. do the following for E number of epochs/iterations:
 - 3.a: go through all N entries of the dataset, assigning each entry X to a cluster:
 - 3.a.1: measure the distance between every cluster centroid C , and X
 - 3.a.2: assign X to the cluster whose distance from X to C is the smallest
 - 3.b: recalculate each cluster centroid C by taking the average value of all entries in that cluster
4. destandardise dataset features and cluster centroids back to the original distribution
5. done

Notably, step 3.a.1 is where the crucial measurement of ‘belonging to a group’ is made, a measurement that is nearly always implemented using Euclidean distance in a Cartesian grid. This observation is a small example of a much larger phenomenon within computational (and more generally, digital) music and art, where Euclidean perspective is pervasive as an assumption of how space is conceived. As artists Jara Rocha and Femke Snelting put it, ‘the obedient adherence to Euclidean perspective ... excels at performing exclusionary boundaries on-the-fly’ (Rocha and Snelting 2018: para. 2). Live coders Kate Sicchio and David Ogborn have also addressed the hegemony of Euclidean thinking about space in software, and are developing a graphical live coding environment to explicitly subvert what they call ‘the dominance of Cartesian representations in 3D graphics’ (Dhaliwal et al. 2022). By assuming a Euclidean space, other possibilities of understanding space are excluded – which includes the relational space of anatomical systems in the body that we are interested in, a point that is addressed directly through dialogue during *AoI* performances.

The following command, when executed by the performers, starts the K-means algorithm running

within the Networked Theatre in each audience member's web browser. This particular invocation runs K-means with eight clusters over 12 iterations, and requests that the algorithm measure similarity between data points in relation to the features 'body', 'perfection', 'gesture' and 'cut':

```
theatre.cluster(8, ['body', 'perfection', 'gesture',
  'cut'], 12, true);
```

As the clustering runs, each iteration brings the process one step closer to an optimal mathematical division of entries into clusters. Each step of the algorithm is displayed within an HTML web element inside the Networked Theatre:

ITERATION 1 of 12 START

```
INITIALIZE CENTROID 1 with features
  0.6581, -0.7868, 0.3359, -0.6956
INITIALIZE CENTROID 2 with features
  0.3291, 1.2922, 1.3589, 0.0679
INITIALIZE CENTROID 3 with features
  -1.2685, -1.4811, 0.1897, -0.6956
INITIALIZE CENTROID 4 with features
  0.3347, -0.4011, -0.2486, -0.5047
measured distance: 1.53051 from hacking-elegant-
anatomy to centroid 1
considering hacking-elegant-anatomy for cluster 1
measured distance: 1.53198 from hacking-elegant-
anatomy to centroid 2
measured distance: 1.25880 from hacking-elegant-
anatomy to centroid 3
considering hacking-elegant-anatomy for cluster 3
measured distance: 2.05031 from hacking-elegant-
anatomy to centroid 4
deciding hacking-elegant-anatomy is in cluster 3
```

5. IN THE PRESENCE OF THE ALGORITHM

AoI relies on three sound strategies in performance: spoken word narratives that guide the audience and provide context, 'sonic marking' (where sonic gestures are attached to steps of the algorithm) and visceral sound design whose goal is to create a highly physical experience of sound. The embodied dimensions of sound are critical to making the algorithm *felt*. From the feelings of vibrations to the material properties of the inflatables and the visual stimuli, audience members have mentioned the impact of the embodied nature and immersive aspect of the performance. One audience member referred to the algorithm unfolding as 'a journey' and as a 'shared discovery' of connections being drawn through the sound and visuals. The feeling of being 'behind the scenes of the algorithm' was also mentioned, which reflects well on our effort to open up the algorithmic process for the audience to experience.

The theatre critic Anna Monteverdi, in a review of an online performance, describes the experience as 'an

interactive storytelling made up of ancient iconographies on the theme of anatomy and knowledge, graphically displayed by an infinite network of strings of codes and connections' (Monteverdi 2023) and confirms the involvement of 'all the senses' while performers compose sound and imagery through code (Monteverdi 2020). Other audience members have described the experience of the performance as 'being awash in an overwhelming volume of human knowledge', referring to a sense of vertigo being exposed to the rapid information processing of K-means. While one audience member who was present in-situ describes the 'childish glee' of trying to keep themselves in balance on the inflatable sculptures while being 'enveloped by the algorithm'.

5.1. An anatomist says ...

Voice plays an especially important role in *AoI*. In much the same way a demonstrator in the eighteenth-century *theatrum anatomicum* would announce the physiological systems, dissective acts and organs on display, the performers in *AoI* guide the attention of the audience to algorithmic systems, live coded gestures and dataset entries. Vocal narratives directly guide the audience to self-reflect on the presence of their bodies and the bodies of those around them (Figure 6). We do this through short vocal intermez-zos called 'anatomic journeys'. In these shared moments, the performers lead the audience's attention away from the visual projections, asking them to focus on sonic textures, rhythms and to oriented themselves to the locations of parts within their bodies in relation to spatial arrangements of data within the geometry of the machine learning algorithm. The audience in situ is also invited to focus on their sense of proprioception and balance while engaging with the inflatable sculptures. The dialogues used in these sections draw attention to the site of the body, inviting the kind of deep breathing that is fundamental to bodily awareness, mindfulness practices and many reflective and contemplative traditions (Tapparo and Zappi 2022).

Voice narration is also a technique commonly used in somaesthetic design projects (Hook 2018), for the audience to turn their attention from one aspect of body experience to another. These vocal intermezzos set the mood for the performance and ask audience members to take an active role in exploring micro-movements within their bodies and to maintain somatic awareness throughout. In contrast, at times when the algorithm is at speed and demands unwavering attention, voice becomes a textural element, processed in real-time through granular live sampling in the browser, creating an ambient bed of vocal texture for the algorithmic rhythms to float on top of.



Figure 6. Audience being guided in an ‘anatomic journey’ by Joana Chicau. Photo: Fenna De Jong.

Take a deep breath.

Breathing is the bridge between the voluntary and involuntary — the sympathetic and the parasympathetic nervous system; the conscious and the unconscious; the inner and the outer.

“Inhale (contract your abdomen) > Retain for 3 to 5 seconds > Exhale (expand your abdomen).

A proximal organ is the nearest organ to another one. Inter- (from Latin inter, meaning ‘between’): between two other structures, such as the inter-costal muscles, running between your ribs.

5.2. The sonic scalpel

Our performance compositions are semi-improvised, involving pre-written snippets of code that are arranged beforehand in the performers’ code editors, while moment-to-moment decisions to explore and sonify sections of K-means are decided on-the-fly. Because an experimental investigation of the algorithm is the goal of dissection, musical rhythms emerge out of the exploratory process, and the creation of specific rhythms – whether genre or culture-specific – is not a consideration. This approach echoes the attitude towards algorithmic art embodied by the ALMAT project, whose aim was to explore musical forms where the algorithm’s compositional authority is so strong

that it effectively ‘becomes an organising principle under which other boundaries such as distinct formats or genres lose relevance’.¹³

Our K-means implementation offers nine steps which can be sonified:

<i>standardize</i>	entries or centroids, standardized feature values
<i>initCentroid</i>	cluster number, feature values of new centroid
<i>iterationStart</i>	iteration number
<i>distance</i>	dataset entry, cluster number, distance measurement
<i>reconsider</i>	dataset entry, new closest cluster, distance measurement
<i>decide</i>	dataset entry, assigned cluster, distance between entry & cluster centroid
<i>computeCentroid</i>	calculate new centroid or keep old, cluster number, centroid feature values centroid feature values
<i>iterationEnd</i>	iteration number
<i>destandardize</i>	centroids or entries, includes cluster number of entries
<i>done</i>	

¹³iem.at. 2017–2020. Aims and questions. <https://almat.iem.at/> (accessed 17 January 2023).

This can be thought of as a ‘timeline’ for K-means. In this way the algorithm becomes a flexible temporal scaffold for musical patterns to unfold, similar to how clave rhythms create a foundational timeline in West African, Brazilian and Cuban musical styles (Toussaint 2002). However, in this case, the timeline is not fixed throughout the duration of the performance, as K-means produces a dynamically changing timeline whose structure changes continuously depending on the initial conditions of the algorithm and the intricacies of the dataset itself. Each of these points can be marked with a sonic gesture as well as given a unique temporal duration on-the-fly by the performers, providing the main compositional tool used for drawing attention to and making felt algorithmic processes in the dissective method.

In live coding it is commonplace to create algorithmic patterns that unfold in time as part of their functional necessity: to make music. However, in the broader scope of computing, algorithms are generally expected to run *as fast as possible*, with a temporality varying by the speed of the computing hardware and the task switching ability of the operating system, at a speed completely imperceptible to humans: a time scale Timothy Barker calls ‘algorithmic micro-temporality’ (Barker 2012) and that Shintaro Miyazaki leverages in his ‘algorhythmic analysis’ (Miyazaki 2012). K-means, and most ready-made algorithms, are generally not written in such a way as to unfold at human-perceptible timescales. We subvert this convention by implementing a temporal intervention, where a musical duration is added to each step of the algorithm together with a ‘hook’ to attach sonic and visual gestures. This allows the performers to temporally zoom in and out of the algorithm in a similar way that a tempo parameter is used in many live coding languages. Our toolkit includes a function named ‘tempo’ in reference to this tradition:

```
tempo(128);
tempo(64);
tempo(32);
tempo(16);
tempo(8);
```

The following is a template example of how an audiovisual gesture, in the form of a self-contained JavaScript function, can be used to ‘mark’ a step of the K-means algorithm. Such a function includes an ‘info’ parameter containing step-relevant information that can be used to vary the audiovisual results, and ends with a return command declaring the step’s durational value as a fraction of the global tempo.

```
KMEANS_STEP.distance = (info)=> {
// sonic and visual manipulations of the theatre
go here
```

```
return 1/2; // duration unit
};
```

5.3. Viscerality: sounds of elegance and disgust

Sound design in AoI is inspired by the irreconcilability of elegance and disgust as found in the eighteenth-century anatomical preparations. In contrast with the quantised rhythms of the algorithmic timeline, we layer a visceral soundscape invoking physiological reactions: high frequency scratching, recordings of viscous fluids lurching through carcasses, the crunching of bones, and the cracking of wood. This sound design is based upon psychoacoustic qualities known to induce bodily effects (Koumura, Nakatani, Liao and Kondo 2021). These ‘visceral’ effects include the tingling experiences of frisson or autonomous sensory meridian response (ASMR), as well as misophonic experiences such as those described by the Spanish word *grima*, the extreme discomfort experienced when hearing nails scratching on a black board. It is interesting to note that within certain Spanish dialects, *grima* is even used interchangeably with *asco*, the word for ‘disgusting’ (Schweiger, Fernández-Dols, Gollwitzer and Keil 2017).

The contrast of elegance and disgust is strongest during moments of close inspection into a single data point, such as when ‘outliers’, those entries that do not fit neatly into any cluster, are examined. At the times when the algorithmic tempo is slowed to less than 10 beats per minute, these misophonic and frisson-inducing sounds are stretched into ambient textures, morphing continuously across the elongated duration of a single algorithmic step. As the performance progresses through the various stages of the algorithm, unfolding sonically and visually, the performers’ voices are transformed with a similar physiological sensibility, by using distortion, filtering and granular techniques to tease out ASMR-inducing whispers and *grima*-inducing screams. The general sonic instability, between elegance and disgust, is mirrored through an embodied instability in the audience, who stretch and rub against the elastic latex rubber of the inflatable sculptures while attempting to maintain balance (Figure 7).

6. REFLECTION AND CONCLUSION

At this point the reader should be familiar with the dissective live coding approach of *AoI*. We now take a moment to reflect on the more general lessons that arise from the project. The foremost take-away being that algorithms used in AI, but more generally in computer science and software engineering at large, need not be seen purely as utilitarian objects. In fact,



Figure 7. Audience laying down in the inflatable sculptures created by Dominique Savitri Bonarjee. Photo: Fenna De Jong.

they offer a world of potential musical and visual forms by making present and sensible their patterns, rhythms and evolution over time. Shintaro Miyazaki and colleagues' earlier work in the Institute for Algorhythmics used basic search and sort algorithms as the material for sonification, creating pieces of audiovisual music. However, unlike the algorithms of Miyazaki's work, contemporary machine learning algorithms and datasets are far more complex and resist such simple sonification approaches. We believe they can be explored successfully instead through a dissective approach involving on-the-fly compositional/improvisational processes that focus selectively on particular algorithmic steps, specific subsections and projections of datasets, or subnetworks of large statistical models. This approach has allowed us to leverage the complexity of such an algorithm into dramaturgical performance narratives. Accessing these underlying structures is not necessarily a straightforward process, and off-the-shelf machine learning libraries do not offer easy ways to create such sonifications, but we could imagine future live coding platforms where algorithms from research and industry are made available as extended options for temporal structures, in the same way that many complex

pattern algorithms are made available in live coding languages and music softwares.

While appropriating and dissecting ready-made algorithms offers a wide breadth of potential musical and artistic expression, *AoI* is also created as a critical reflection on prevailing tech cultures of data capture and analysis. We hope that our approach inspires digital artists to look beyond the excitement around new technical tools and AI capabilities to consider how the experiences they create encourage critical reflection and subvert largely unchallenged narratives around technology. By making felt and sensible algorithms that are used in everyday digital technologies, artists may enable access to them that is potentially revelatory, opening up venues for both understanding of what algorithms are as well as what they are not. In this respect, live coding provides a unique way in, and makes possible aesthetic investigation algorithms at various levels of technical depth in a flexible and open way.

In conclusion, the anatomical perspective pursued in *AoI* encourages an understanding of computing systems reflected through physiological bodies, which opened novel possibilities for exploring embodied experience in live coding performance. In the future we

would like to further open up the Networked Theatre with detailed documentation so that others can use it more easily, either in workshops or as a performance and exploration tool. We also look forward to exploring other algorithms and datasets, and how they might be understood differently through the language of anatomy.

Acknowledgements

The ongoing development of Anatomies of Intelligence has been made possible with financial support from many cultural organisations including V2_Lab for Unstable Media, Umanesimo Artificiale, iii (Instrument Inventors Initiative) and STROOM Den Haag. Additionally, this article was supported by the Arts and Humanities Research Council Doctoral Training Partnership (Techne AHRC/UKRI) [grant number AH/R01275X/1].

We also send a special thank you to Dr Marieke Hendriksen for illuminating conversations on the history of European anatomical science.

DATA AVAILABILITY STATEMENT

The dataset used in this work is composed from images and text drawn from public/open sources, and can be accessed via the github repository: <https://github.com/anatomiesofintelligence/dataset>.

REFERENCES

- Akten, M. 2015. Data Dramatization. *Medium.com*. <https://memoakten.medium.com/data-dramatization-fe04a57530e4> (accessed 12 September 2022).
- Baalman, M. A. 2015. Embodiment of Code. *Proceedings of the First International Conference on Live Coding*, ICSRIM, University of Leeds, 35–40. <https://doi.org/10.5281/zenodo.18748>.
- Barker, T. 2012. *Time and the Digital*. Chicago: University of Chicago Press.
- Bernardo, F., Kiefer, C. and Magnusson, T. 2020. Designing for a Pluralist and User-Friendly Live Code Language Ecosystem with Sema. *5th International Conference on Live Coding*, University of Limerick, Limerick, Ireland.
- Blackwell, A., Cocker, E., Cox, G., McLean, A. and Magnusson, T. 2022. *Live Coding: A User's Manual*. Cambridge, MA: MIT Press.
- Chatzichristodoulou, M. and Zerihan, R. (eds.) 2012. *Viscerality in Performance: Intimacy Across Visceral and Digital Performance*. London: Palgrave Macmillan.
- Chicau, J. 2017. A WebPage in Two Acts. *Proceedings of xCoAx the 9th Conference on Computation, Communication, Aesthetics & X*, Lisbon, Portugal.
- Cocker, E. 2016. Performing Thinking in Action: The Meletē of Live Coding. *International Journal of Performance Arts and Digital Media* 12(2): 102–16. <https://doi.org/10.1080/14794713.2016.1227597>.
- Collins, N. 2011. Live Coding of Consequence. *Leonardo* 44(3): 207–11. https://doi.org/10.1162/LEON_a_00164.
- Cox, G. and Soon, W. 2021. *Aesthetic Programming: A Handbook of Software Studies*. London: Open Humanites Press.
- Cunningham, A. (ed.) 1993. *English Manuscripts of Francis Glisson*. Vol. 1: *From 'Anatomia Hepatis', 1654*. Cambridge: Cambridge Wellcome Unit.
- Dhaliwal, A., Ogborn, D., Kim, E., Sicchio K., Hinic, K., Ahmed, S., et al. 2022. LocoMotion: Live Coding 3D Movement on the Web. *MoCo'22: 8th International Conference on Movement and Computing*, Chicago, IL, USA.
- D'Ignazio, C. and Klein, L. F. 2020. *Data Feminism*. Cambridge, MA: MIT Press.
- Dourish, P. 2001. *Where the Action Is: The Foundations of Embodied Interaction*. Cambridge, MA: MIT Press.
- Ghajargar, M., Bardzell J., Smith-Renner A. M., Höök K. and Krogh P. G. 2022. Graspable AI: Physical Forms as Explanation Modality for Explainable AI. *TEI '22: Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction*. Daejeon Republic of Korea: ACM: 1–4. <https://doi.org/10.1145/3490149.3503666>.
- Han, F. and Reus, J. 2018. Perform_Tech Conversation Series: Jonathan Reus. www.composerfh.com/perform-tech (accessed 17 July 2023).
- Hemment, D., Aylett R., Belle V., Murray-Rust D. and Luger E. 2019. Experiential AI. *AI Matters* 5(1): 25–31. <https://doi.org/10.1145/3320254.3320264>.
- Hendriksen, M. 2015. *Elegant Anatomy: The Eighteenth-Century Leiden Anatomical Collections*. Leiden: Brill.
- Hoogland, T. 2019. Mercury: A Live Coding Environment Focussed on Quick Expression for Composing, Performing and Communicating. *Proceedings of the Fourth International Conference on Live Coding*, Madrid, 353–64.
- Hook, K. 2018. *Designing with the Body: Somaesthetic Interaction Design*. Cambridge, MA: MIT Press.
- Knoeff, R. and Zwijnenberg, R. 2015. *The Fate of Anatomical Collections*. Farnham, UK: Ashgate.
- Koumura, T., Nakatani, M., Liao, H. I., Kondo, H. M. 2021. Dark, loud, and Compact Sounds Induce Frisson. *Quarterly Journal of Experimental Psychology* 74(6): 1140–52.
- Lakoff, G. and Johnson, M. 2008. *Metaphors We Live By*. Chicago: University of Chicago Press.
- Machon, J. 2009. *(Syn)Aesthetics: Redefining Visceral Performance*. London: Palgrave Macmillan.
- Miyazaki, S. 2012. Algorhythmic: Understanding Micro-Temporality in Computational Cultures. *Computational Culture* 2. <http://computationalculture.net/algorhythmic-understanding-micro-temporality-in-computational-cultures>.
- Monteverdi, A. 2020. Il teatro dell'algoritmo: Umanesimo Artificiale per le Residenze Digitali – Digital Performance. www.annamonteverdi.it/digital/il-teatro-dellalgoritmo-umanesimo-artificiale-per-le-residenze-digitali/ (accessed 17 January 2023).
- Monteverdi, A. 2023. Residenze Digitali: lo spettacolo si fa on line. *Umanistica Digitale* 15 (July): 151–67. <https://doi.org/10.6092/issn.2532-8816/16888>.

- Norman, S. J. 2016. Senses of Liveness for Digital Times. *IETM Publications*. <http://sro.sussex.ac.uk/id/eprint/61175/>.
- Penney, M. 2016. www.sessions.edu/notes-on-design/all-about-glitch-pixelsorting/ (accessed: 12 September 2022).
- Pirró, D. and Rutz, H. 2022. ALMAT – Continuous Exposition, Research Catalogue. www.researchcatalogue.net/view/381565/381566/0/0 (accessed: 17 January 2023).
- Reus, J. 2021. Vacuum Forms. In *Algorithmische Segmente | Algorithmic Segments*. Graz: Reagenz Verlag.
- Rocha, J. and Snelting, F. 2018. Xyz. *Fictional Journal*. www.fictional-journal.com/xyz/ (accessed 9 August 2022).
- Schweiger Gallo, I., Fernández-Dols, J. M., Gollwitzer, P. M. and Keil, A. 2017. Grima: A Distinct Emotion Concept? *Frontiers in Psychology* 8. www.frontiersin.org/articles/10.3389/fpsyg.2017.00131.
- Tapparo, C. S. and Zappi, V. 2022. Bodily Awareness through NIMes: Deautomatising Music Making Processes. *NIME*. <https://doi.org/10.21428/92fbeb44.7e04cfc8>.
- Toussaint, G. 2002. A Mathematical Analysis of African, Brazilian, and Cuban *Clave* Rhythms. www.semantic-scholar.org/paper/A-Mathematical-Analysis-of-African%2C-Brazilian%2C-and-Toussaint/0bc571d9fac5f3eaaa0c733eb6e70aa0536bb977 (accessed 17 July 2023).
- Tremblay, P. A., Green, O., Roma, G., Bradbury, J., Moore, T., Hart, J. and Harker, A., 2022. The Fluid Corpus Manipulation Toolbox (v.1). *Zenodo*. <https://doi.org/10.5281/zenodo.6834643>.
- Xambó, A., 2021. Virtual Agents in Live Coding: A Short Review. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2106.14835>
- Ward, A., Rohrhuber, J., Olofsson, F., McLean, A., Griffiths, D., Collins, N. and Alexander, A. 2004. Live Algorithm Programming and a Temporary Organisation for its Promotion. *Proceedings of the README Software Art Conference*, 289: 290.