Convolving Signals
Thinking the performance of computational processes

Andrew Murphie

Figure 1—Musician Loscil Performing. image Courtesy of Kelly Hailey/CIFF. [1]

> … a machine is not a living organism, and therein lies its force. (Stiegler, n.d.)

> A fundamental operation in signal processing, convolution "marries" two signals. (Rabiner and Gold, 1975; in Roads, 2004: 209-210)

In contemporary performance, "action" seems to include acts, emergent action and the potential for action—even sometimes non-action. Performers include the human and nonhuman, the living and the nonliving. Through all this runs a complexity of technics—technologies and techniques—and these increasingly involve computational processes. How do we think the last of these—computational processes in performance? [2] This article gives some answers to this question by engaging with four different but overlapping examples of the way that computational processes are brought into performance: the digital signal processing technique of convolution, live coding as performance, the electronic music of Loscil and Stephan Mathieu, and Sher Doruff's work with the collaborative network performance software, *Keyworx*. It suggests the necessity of understanding the imperceptible aspects of computational processes, such as signal processing, as performing, even if not always directly presented. This in turn suggests a more subtle understanding of the micro-dynamics of performance in general. As part of this, I propose that more attention is paid to the transformation of signals of all kinds, in and perhaps even as performance. The article concludes by proposing that we now live, at least in part, in a culture of ongoing convolution of signals. This is a culture more self-consciously immersed in the ongoing differentiation (and integration) of intensities. The article therefore raises questions about the general performativity of a culture and technics increasingly enmeshed with computational processes.

Thinking Performance and Computational Processes [3]

Even the ancients understood the role of machines in performance and we have more recently become accustomed to thinking about media such as video or even multimedia in performance. However, we are perhaps less accustomed to thinking of less visible technical processes such as computational or signal processes as part of whatever it is we conceive action to be. If key aspects of culture are now becoming *computational*—a matter of the often imperceptible performance of algorithms and signal processing—how are we to think about action today?

This article begins by discussing an important technique in digital signal processing—convolution. Convolution is the folding of signals into each other (enabled by the mathematical convolution of functions). It is found in sound composition and performance but, as we shall see, in many other areas as well. Convolution suggests an understanding of the performativity of computational processes in terms of 'signaletic materials' rather than (initially at least) in terms of the transmission of signs, messages and narrative (Thomsen, 2012). [4]

Thinking with convolution suggests that we have much to gain by thinking performance more generally in terms of signal flows and breaks, signal events and signal work, relations and varying intensities. Thinking performance in terms of signal (and the various technics by which signal is processed, including computational processes) allows for a more open concept of the "liveness" of performance. [5] I will simply propose here that liveness arises from any event in performance that gives the *sense* of life. That is perhaps enough. If so, any sense of life is assembled immanently, often by events of signal work. One might say that liveness is the register of ongoing work—signal and other work—in creating, combining and fragmenting signal and other relations. This is work in the distribution of intensities, felt and otherwise. Previously I have suggested that performance's 'first impulse is towards … a multiplication of intensities between intensities, differences between differences', and that 'all performance acts fold into, or unfold, the world of moving intensities'. As such, all performance events balance or distribute intensities in their own way (Murphie, 2009: 222). However, in proposing that a sense of life arises from this differential distribution of intensities, I am not proposing a challenged ('it's us versus technology!') or extended ('artificial life!') vitalism. In fact, the *sense* of life of performance does not need life per se. We can instead begin with the way that every new differential distribution in performance involves a mix of signals and relations of all kinds, in process. From this, various kinds of sense, including that of life, arise (and into this they fall). This perspective allows us to consider a kind of anti-structural and dynamic organisation of performance made more complex in performance's engagement with technics. Performance here can be conceived as work with flows, including but not limited to, signal flows of all kinds.

We might say that performance has always been a mix of forces of the 'living' and the 'dead' (Stiegler, n.d.). Or; we might say performance is a kind of cross-signal processing of 'living' and the 'dead' as signaletic materials or forces. Only secondarily, and only immanently, does any sense of life emerge. So for vitalism, liveness or, crucially, even presence, we might substitute something like an ongoing mix of the *tendencies* that generate signals, and tendencies that are then carried as literal in*form*ation by signals. [6] As Brian Massumi writes, 'there is no need to posit a life-substance or life-force "behind" appearances. All you need to posit is the appearance of a tendency' (2008: 38-39). I will discuss convolution in some detail here in order to approach the broader question of the performativity *of* computational processes, as part of thinking about performance *and* computational processes. To think with convolution and other forms of signal processing, in the like of live coding, is to think with immediate tendencies and intensities, rather than, initially at least, with liveness, presence or narrative, message or theme. The performance of laptop music or dance and technology performances (for example, Chunky Move's *Mortal Engine*) [7] would be two examples of work with signal tendencies and intensities that lend themselves to such thinking, although the concept of signal transformation can be taken very broadly (Murphie, 2012).

In such contexts, do computational processes such as convolution perform, or do they merely supplement performance? What might thinking about the performance of computational processes themselves in, for example, live coding, tell us about performing-with them? If 'a machine is not a living organism' but 'therein lies its force' (Stiegler, n.d.) what does the mix of different forces in performance with technologies tell us about the 'nature of performance itself' (Murphie, 1990: 209)?

Recent technological interventions—computational processes and media technologies in general—have undoubtedly changed performance. For one thing, all technologies demand new ways of 'negotiating presence' (Murphie, 1990). Indeed, as I have begun to suggest above, presence, often seen as the bedrock of performance, can now be understood differently, as a differential distribution (Murphie, 1990 & 2010; Cull, 2012: 79 & 189). Technical interventions also seem ambivalent with regard to the body, particularly the forces and movements of the body in performance (and to related questions, for example of 'intentionality'). There are often new possibilities for the body in performance, but it can also be undermined, for example through an overcoding of gesture (Manning, 2006). Again, there is perhaps a new kind of differential distribution of forces and movements involved. This is a differential distribution that itself differentiates over time. The result can be an ongoing, symbiotic transformation of both body and technics. As Anna Munster puts it:

> ... the incorporeal vectors of digital information draw out the capacities of our bodies to become other than matter conceived as a mere vessel for consciousness ... we may conceive of these experiences as a new territory made possible by the fact that our bodies are immanently open to these kinds of technically symbiotic transformations. (2011: 19) [8]

When the technical processes involved are largely imperceptible, as they are with computational processes, the register of all these questions expands. This register is not always easy to perceive. The processes are hidden; literally micro-process*es* of microprocess*ors*. So these are micro-performances, or, better, we are dealing with a multiplicity of performances, and the resonances of patterns of relation, that are able to scale across micro and macro.

In sum, understanding how computational processes and what I call 'signal work' themselves perform allows us to think a little differently about other aspects of performance—bodies, life, forces and movement and other more easily perceived elements of performances, including more visible technical elements (lighting, screens, sound, or even robotics). All of this resonates with the broader performativity in which we all live in a general performance culture.

But let us begin with convolution.

Convolution

Convolution would not be possible without the discoveries of Joseph Fourier in the early 19th century (although the mathematics was beginning to be developed before him). Fourier was a mathematician who in the time of Napoleon suggested the possibility of a greenhouse effect. He also developed a mathematics of heat, light and sound (among other things) that enabled much of what we experience as electronic sound today. Fourier was personally attracted to the kind of heat he first discovered in the Egyptian desert while serving in Napoleon's army, to the extent that he thereafter kept himself as hot as possible as much as he could. [9]

Fourier realised that all the phenomena which could be understood as wave events—again heat, light, sound but in fact all events of signal—have thoroughly complex waveforms, with for example, a complex mix of frequencies. However, he also discovered that these could be broken down into a series of co-mingled, far simpler waves, sine waves. Recombined they would produce the initial sound (or combined differently a different sound). In sound, a sine wave is the purest sound, and as Brian Eno has remarked also the most boring sound, consisting of the kind of regular wave a child might draw, for a sound that has no harmonics (BBC, n.d.). Of course there is still an infinity of possible sine waves, but they are the crucial elements of a kind of infinite regularity that can be found, and therefore worked with, mathematically, in the structure of variation that are more complex wave events such as sound. [10] Fourier provided the means of breaking signals down into simple components that can then be recombined differently. He also provided a way of mathematically analysing and transforming aspects of these waves, in what are called Fourier transforms. Faster computers and better processes have more recently allowed for what are called fast Fourier transforms, which allow this processing to be done in real time (and thus changing sounds in real-time). [11]

Much was made possible by Fourier's discoveries. For example, spectrum (or spectral) analysis allows the analysis of a soundwave (or other wave), its breakdown into components and the recombining of them so that for example, a violin's soundwave could end up sounding like a tuba. Between spectral analysis and processes such as convolution an infinite variety of ways of working sound becomes possible. Those who work with such processes usually say that such sounds have a life of their own. Such processes have lead to a series of practices, and understandings, based on the fact that there are an infinite series of ways of extending performative action through all the potentials and intensities assembled in a performative event.

As just suggested, this is not so much a matter of vitalism as a question of a complex work with tendency. This is in turn a matter of ongoing process. It's not called signal *processing* for nothing. Massumi writes, 'A tendency, as it appears, is always, only, and entirely in process' (2008: 39). Moreover, here we are talking about processes of fragmentation and recombination—a breakup, potentialising and recombination of tendencies. Adrian Mackenzie writes that 'Fourier analysis and the Fourier transform express the waveform as a set of partial tendencies (partial differentials) rather than a single tendency' (76).

Based on these fundamentals then, we can consider convolution in more detail. For the engineer, convolution—

> … is a mathematical way of combining two signals to form a third signal. It is the single most important technique in Digital Signal Processing. [In a prevalent use of convolution], using the strategy of impulse decomposition, systems are described by a signal called the impulse

> response. Convolution is important because it relates the three signals of interest: the input signal, the output signal, and the impulse response. (Smith, 1997: 107)

Technically, a ‘convolution is an integral that expresses the amount of overlap of one function $_g$ as it is shifted over another function $_f$’ (Wolfram Mathworld, n.d.). In digital signal processing, this could involve a short arbitrary input into an established system (introducing for example, a certain kind of resonance into the sound of a digital musical instrument), but it could just as easily involve the convolution of two fuller impulses or signals. Convolution is sometimes called faltung (German for “folding”)’ (Zhen, 1990: 288). In fact, Waigai Zhen points to four different processes involved in convolution: folding, translating, multiplying and integrating (1990: 288). Convolution is in all this a relational engagement between different functions over time, with uses in, among other areas ‘statistics, computer vision, image and signal processing, electrical engineering, and differential equations’. Many processes that can be turned into functions can be processed by anything else that can be. So for example, all the values for a musical sample over time can be used to transform another music sample (in for example echo or reverb).

Convolution makes it possible to utilise the opening up of the immanence of (computational, performance, signal) events. Yet it first needs a subtraction of that which usually—literally—over-codes these events them in terms of, for example, narrative or character or finished composition (or even perhaps liveness or presence) (Bertelsen and Murphie, 2012). Convolution works in partnership with this subtraction. An alternative way of understanding this is that convolution undoes over-coding via a precise technical form of *non-bifurcation, or undoing of bifurcation*. [12] It is literally a folding of sound samples or other pieces of code/signal (along with accompanying abstractions) into each other. This folding can be partial. In sound, for example, samples can have aspects of their shapes, frequencies, amplitudes, their intensities, in fact any aspect, convolved. This applies also to images or in principle anything that can be digitised. In fact, in principle it is a method that can easily go beyond the digital, electronics or even technologies to cultural techniques. At the most general level, convolution is a process of infolding active contrast. As such, convolution is a principle/practice of opening up what is already there to its own inherent differential distribution, its own potential for continuous variation. Convolution is therefore an ongoing reworking of potential within the immanence of material dynamics.

As mentioned, in digital signal processing and electronics, convolution is enabled by the mathematical folding into each other of functions and, via a deployment of these functions in a combination of abstract and concrete implementations, a folding into each other of *signals*. [13] In much digital signal processing, convolution is limited to the folding of an ‘impulse response’ or IR, into a input signal to produce a new output. So for example, a digital analysis of the impulse response of a particular environment, such as a concert hall can then be folded into an input signal, for example, a digital recording of a violin. [14] Every micro sample of the violin sound is folded in with the IR to produce a new sound. The result, with which most people will be familiar, is the sound of a violin played in a concert hall. Other common sound transformations—modulation, filters and resonances of many kinds—can be produced via convolution. A similar situation applies in convolving video signal, as can be seen most obviously in VJing (live video mixing). Again however, convolution need not be limited to an IR and an input signal. It can also involve two fuller signals (or extended partial aspects of signals) being folded into each other. Many musicians claim that in this fuller cross-signal processing things get out of control very quickly, with a range of unexpected sounds produced. In addition, there is no reason why it has to be a question of only two signals, although things get very complex once you add more than two into the mix. On the other hand, a signal can be convolved with itself (for example, over time). Not only can different co-existing sounds be convolved, convolution can also convolve (even within the same signal) across time (in, for example, delay effects). In fact, even in its use in everyday wi-fi, convolution usually involves a folding of both space and time at once. What is called deconvolution is an unfolding of this folding (as in the Viterbi algorithm, again used in wi-fi). [15]

Convolution is both a well-known technical problem and solution in the history of working with signals. One way of looking at Shannon and Weaver's famous, and foundational 'Mathematical Model of Communication' from 1949 is that it was not just a matter of a kind of *competition* between signal and noise, with noise to be overcome; rather this involved a *convolution* of signal and noise (the latter really meaning unwanted signal convolution as much as straight interference). However, this convolution could be overcome by work with further convolution. An example is again contemporary wi-fi. Mackenzie points out that convolution is even more crucial once communication goes wireless. In wirelessness, signal has to travel, "by itself" through what is known as an air interface in which it competes and collides, and is folded in with many other signals. The situation becomes even more complex as there is more take up of wireless communications.

From all this we can derive three key points that perhaps go against our usual understanding of the digital as always breaking down events into discrete micro-elements. These favour thinking in terms of micro-*processes* capable of producing multi-scalar events.

First, signal convolution shows us that these breakdowns into elements are arguably a secondary, at best parallel, series of events or tools within a wider computationality. Perhaps as a hangover from thinking about the world in terms of signs, the elements of language and discourse, the digital world is too often thought over-simplistically in terms of zeroes and ones, and not enough in terms of signal processes, functions in operation, folding and unfolding, in short in terms of durations and algorithmic *events*.

Second, as event a signal convolved through time has something like a shifting mood. As Mackenzie puts it with regard to wi-fi—

> Convolution [en]coders take their name from the way they base what they transmit at the current point in time on what has been transmitted earlier. They begin to build a "state of mind" concerning what has preceded the current moment in the data stream: 'We have states of mind and so do encoders. We are depressed one day and perhaps happy the next from the many different states we can be in .. We can say that encoders act this way too. What they output depends on … their state of mind' (Langton, 1999: 3). (Mackenzie, 2011: 79)

Here we find another hint as to what a computational process might bring to a performance event. Simply put, it expresses a pliable 'state of mind', or better mood, or in Whitehead's terms, 'affective tone'. This could be a nonhuman "state of mind". This echoes the idea that signal events are in part —literally, processually—a shifting attunement within a broader event. The 'affective tone is concern—in Whiteheadian terms—for the event itself' (Manning, 2009: 40). In fact, *the event involves an assembling or convolution of such concerns* under the conditioning of a particular perspective. As Whitehead puts it—

> Expression is the gift from the World as many to the World as one. Selection belongs to expression. *A mood of the finite thing conditions the environment.* There is an active entity which fashions its own perspective, implanted on the world around. (1968: 21)

Note that this is a matter of conditioning the environment, not determining it. And note that this allows us to understand signal processing as both an ongoing and singular adaptation to contingency, rather than a simple imposition of “control”, and also a gift to ongoing events of expression.

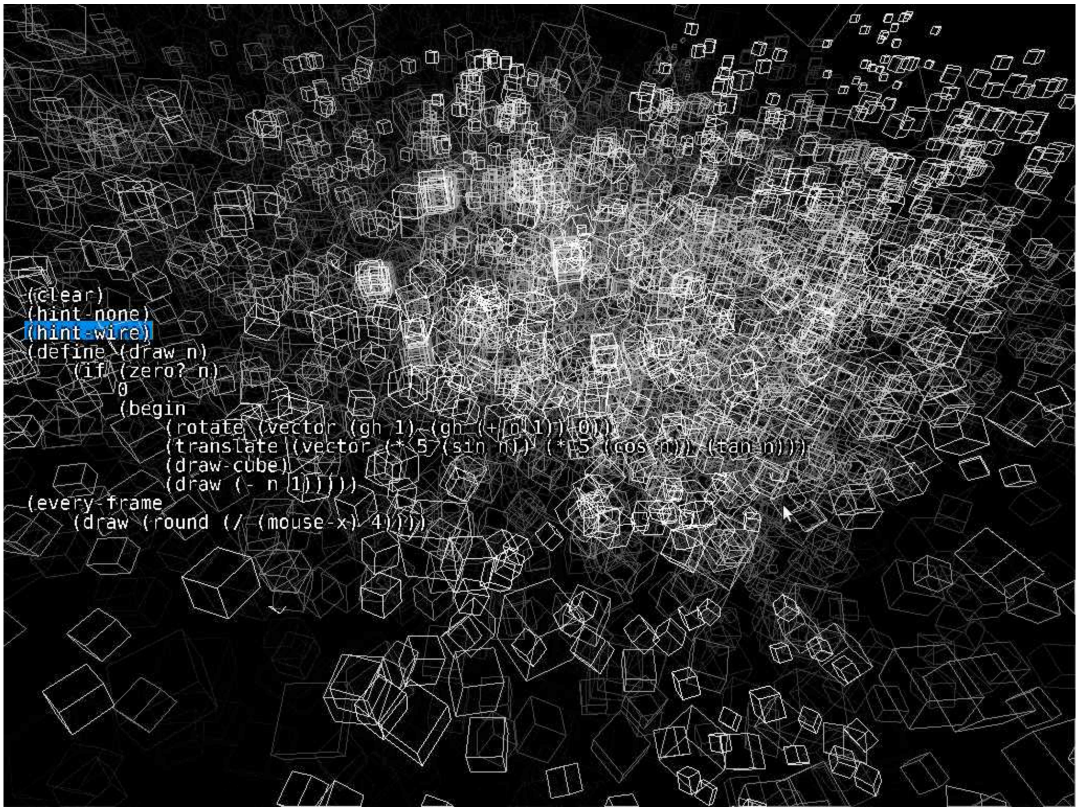
Third, this also works in the other direction, in that engagement with the environment intensifies a specific signal-event. Mackenzie points out that convolution in wirelessness involves an intensification of relations, and intensifications of movements (of signal) in order for a signal to survive. In wi-fi—

> The “convolution” consists in this folding of the data stream to incorporate information about what was transmitted before. Each packet, datagram, or frame represents not just information, but a relation to what came before. … Intensified movements in transmitter and receiver compensate for disturbances or errors arising from what cannot be made fully part of the system: the pathway along which signals propagate. Convolutional codes assume that communication comprises a process of conjunction that cannot be fully controlled or determined and that therefore cannot be definitively transmitted. … Convolutional coding creates a great many relations internal to the signal, so that their relationality exceeds any possible interruption by another signal. (2011: 80-81)

It is necessary to add that individual signal events themselves arise and fade way—in Whitehead’s terms they perish once satisfaction has been achieved. After this, they often remain in mutating/ resonating forms to be picked up by later signal convolutions. In performance, this intensification, signal events arising and perishing, can involve ongoing convolution with other similar signals, other kinds of signals, other intensifications of relations and movements. It is this that provides the substance of the ongoing individuation and trans-individuation, in Simondon’s terms, of performance.

I can now move on from the general consideration of convolution, to some more specific accounts of performance and composition involving computational processes. I will discuss live coding, not in relation to convolution but rather to more general questions about performance and computational processes. I will then turn to examples of convolution in the work of musicians Loscil and Stephan Mathieu. Then I will discuss the software package *Keyworx* as providing a general, networked ‘convolutionary’ performance space—one that is ‘translocal’ (Doruff in Doruff and Murphie: 2012), and one that exemplifies a culture of ongoing convolution.

Live Coding



```
(clear)
(hint-none)
(hint-wire)
(define (draw n)
    (if (zero? n)
        0
        (begin
            (rotate (vector (gh 1) (gh (+ n 1)) 0))
            (translate (vector (* 5 (sin n)) (* 5 (cos n)) (tan n)))
            (draw-cube)
            (draw (- n 1)))))
(every-frame
    (draw (round (/ (mouse-x) 4))))
```

Figure 2—'Live coding practice (Fluxus)'. Image by Akinori Kinoshita. Taken, with permission, from http://akinoshi.blogspot.com.au/2010/09/live-coding-practice-fluxus.html

In *Speaking Code*, Geoff Cox and live coder Alex McLean take a more or less Austinian approach to the question of performance and computational processes. They develop this into a *politics* of the performance of code. They point out that events of code *in action* resemble performative speech acts in that 'they do what they say … at the moment of saying it' (2013: 35). Moreover, in that it is performative, code is not just a matter of the repetition of convention (36). Code therefore carries a kind of charge, an action potential, further complicated and empowered by the complex intra- and extra-relationalities of the computational processes that code enables. This has something of a parallel in the potential for movement of bodies, perhaps best expressed in Erin Manning's concept of 'preacceleration'. Pre-acceleration is the incipience of movement, 'a virtual becoming–a tendency toward movement–through which a displacement takes form' (Manning, 2006, unpaginated). We can well imagine the fields in performance within which the action potential of code might meet the preaccelerations of movement and bodies. Although Manning's preacceleration is certainly not as preformed as code (as much as I argue that code itself might not be totally preformed).

In the context of action, Cox and McLean point out that machines do not 'merely do what they are programmed to do', just as the performative (speech) act 'extends the unstable relation between the activities of writing, compiling, and running of code as a set of interconnected actions (as the practice of live coding demonstrates so well)' (38). Cox goes on to harness Arendt's work on the performative arts to the cause, pointing out that for Arendt, 'the performative arts are particularly resistant to reification as the least materialistic of the arts' (59). All this begins to emphasise how different code looks to us in performance than in our usual more static (we might say, office informed) concept of it.

Here then, the example of live coding can tell us a great deal about all performance involving computational processes.

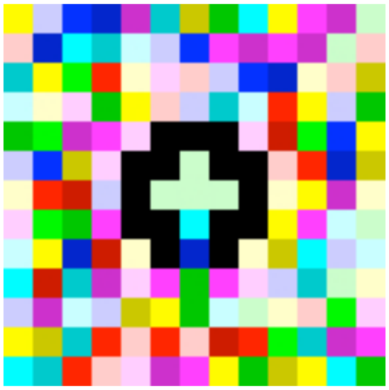
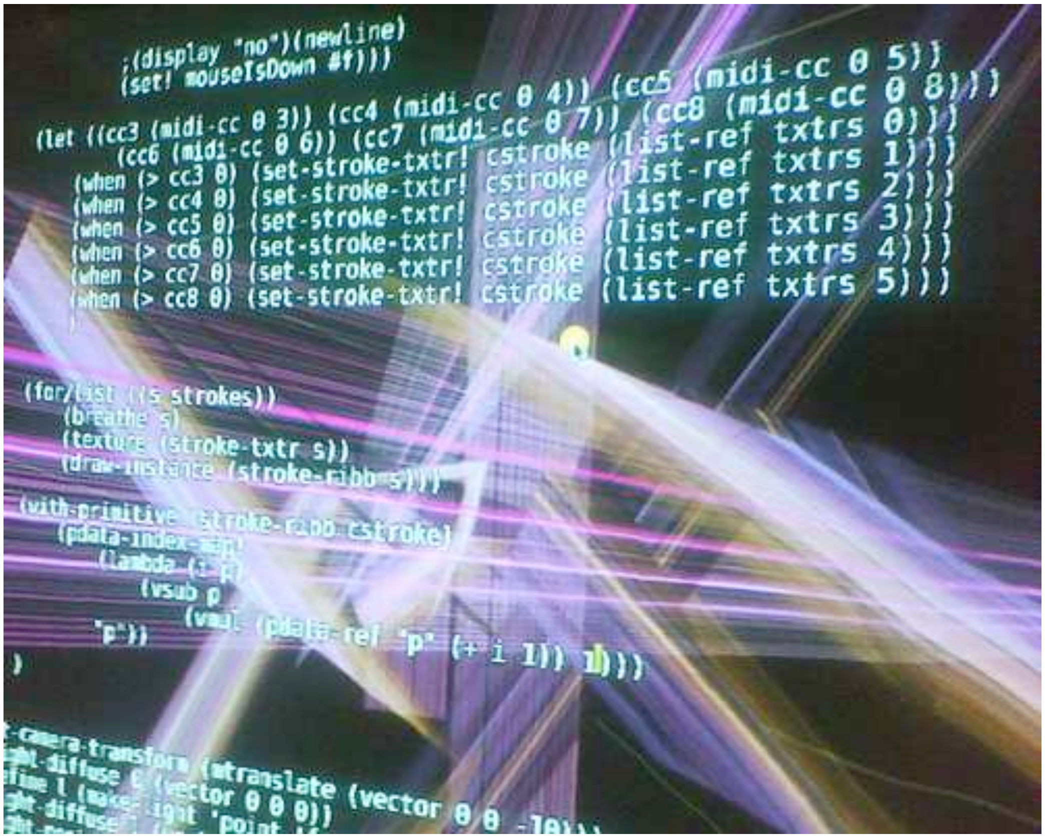


Figure 3—Source code written in the Piet language with two dimensional, colour syntax. Prints out the text "Hello, world!". Image © Thomas Schoch 2006. Used under the Creative Commons BY-SA 2.5 license. (found via Alex McLean, 2010) [16]



```
    ;(display "no")(newline)
    (set! mouseIsDown #f)))

(let ((cc3 (midi-cc 0 3)) (cc4 (midi-cc 0 4)) (cc5 (midi-cc 0 5))
      (cc6 (midi-cc 0 6)) (cc7 (midi-cc 0 7)) (cc8 (midi-cc 0 8)))
   (when (> cc3 0) (set-stroke-txtr! cstroke (list-ref txtrs 0)))
   (when (> cc4 0) (set-stroke-txtr! cstroke (list-ref txtrs 1)))
   (when (> cc5 0) (set-stroke-txtr! cstroke (list-ref txtrs 2)))
   (when (> cc6 0) (set-stroke-txtr! cstroke (list-ref txtrs 3)))
   (when (> cc7 0) (set-stroke-txtr! cstroke (list-ref txtrs 4)))
   (when (> cc8 0) (set-stroke-txtr! cstroke (list-ref txtrs 5)))
   )


(for/list ((s strokes))
    (breathe s)
    (texture (stroke-txtr s))
    (draw-instance (stroke-ribbons)))

(with-primitive (stroke-ribb cstroke)
    (pdata-index-map!
        (lambda (i p)
            (vsub p
                (vmul (pdata-ref "p" (+ i 1)) 1)))
        "p"))
)


-camera-transform (mtranslate (vector 0 0 -10)))
ght-diffuse 0 (vector 0 0 0))
efine l (make-light 'point 'f
ght-diffuse
```

Figure 4—Live Coding (Fluxus). [17] *Image © Peter Coen 2009. Used under the Creative Commons BY 2.0 license. (http://www.flickr.com/photos/angorawol/3556733767/)*

In live coding—

> … programmers make music [or images, or transformations of code itself, etc] in keeping with the expressive qualities of live performance, by coding in real time (using a command line interface for example) ... 'the usual design-implement-test development cycle is discarded in favour of an immediate actions and reactions more suited to creative exploration'. (61) [18]

In that live coding works with a dynamism that is already there in code's potential, it suggests that the conception and implementation of code could always involve a more dynamic set of events than is often acknowledged, either in the understanding of performance, or in the designing of computational processes into multimedia performances. Cox and McLean go on to discuss a text editor developed by McLean called *feedback.pl*. In this text editor the programmer is able to edit the code while the code is running, but the code can also modify itself. As Cox points out, 'self-modifying code blatantly breaks the determinism of code and makes it explicitly performative'. This leads coder and code into what he terms 'an uncertain relation' (61). Cox argues for 'dynamic recombinations of speaking, thinking, encoding', the rejection of 'forms of totality' and the recognition that both 'voice *and* code', and we might add other aspects of performance, are all 'always ready for action and at the same time ready to run out of control (like a live-coding performance)' (109). He suggests that 'both subjectivity and code recursively write their own instrumentation', with both subject and code 'standing between what is possible and what actually exists'.

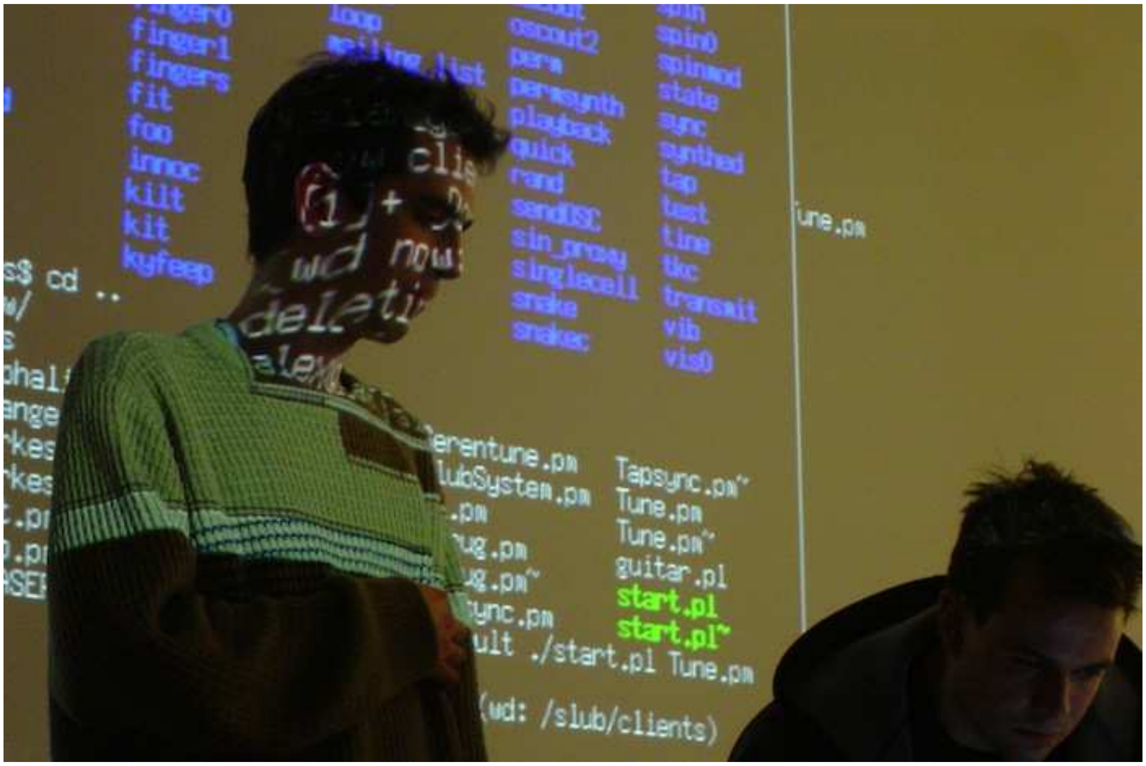


Figure 5—Alex McLean in a Slub performance. [19] *Image by Renate Weiser. Sourced with permission from Alex McLean.*

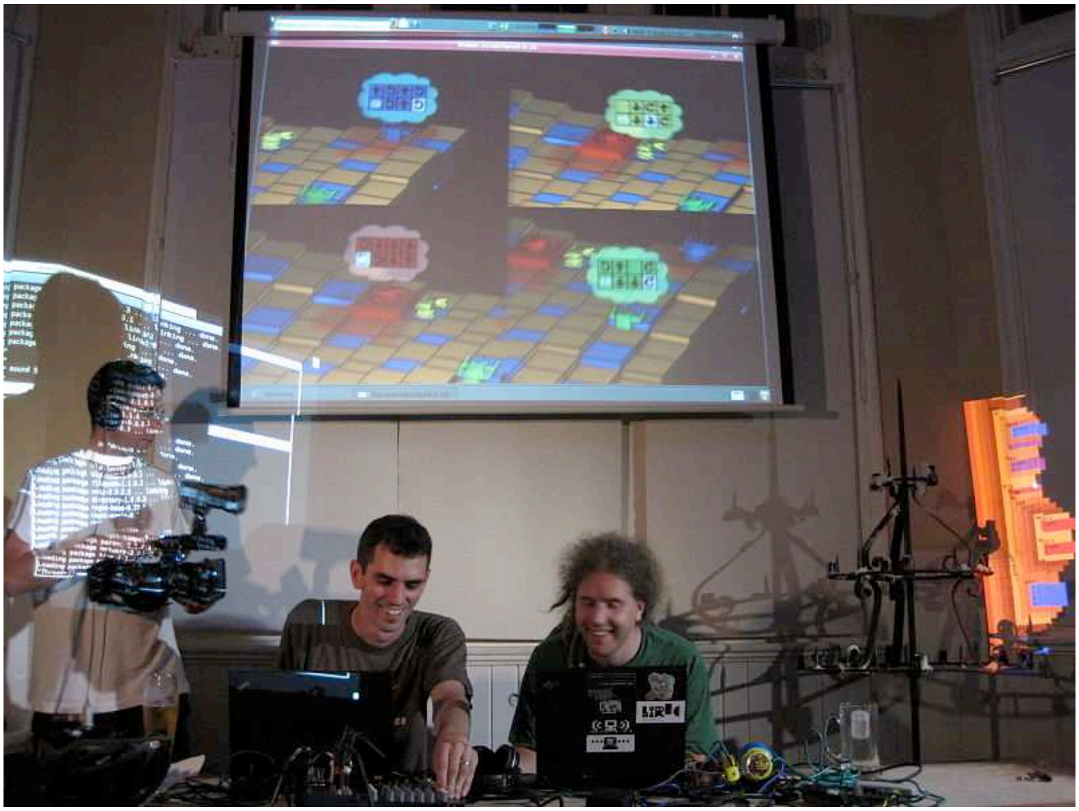


Figure 6— Slub performing live at the Roebuck pub in London. Image by Philippa. Source: http:// en.wikipedia.org/wiki/File:Slub_live_Roebuck.jpg. Used under the Creative Commons BY 2.0 license.

Morten Breinbjerg also sees live coding as a kind of dynamic instrumentation (in the musical sense). Quite differently to commercial, relatively standardised software, for example for music sequencing or even some generative software, live coding emphasises 'the materiality of software and human-computer interaction as a complex and mediated process' (162). This materiality also includes the audience, as in live coding, 'laptop screens [are] projected so that the audience can see the program being written' (164). Breinbjerg points to the liveness, we might say 'eventness of the act' (173), of the engagement with software in live coding. The focus is on 'algorithmic calculations, randomness, generativity and programmability, among other things' (166). In short, 'it is no longer the ability to edit music that is being explored'. It is no longer the given determinations of fairly fixed software imposing themselves on performative interactions that are being explored. Rather, the software, code and laptop, along with projection in the performance space, have become a kind of musical instrument (166). Coding 'becomes an expressive act', one that acknowledges 'the conditions of its own making' (167). Breinbjerg further suggests that this involves much more than the now traditional understanding of composition or performance as what we now call remix (174). More than this, 'in live coding the program is caught into time structures: the time of the program and the time of the process in which the programme is being modified'. This is always a kind of a struggle—a differential intensity that encourages the like of convolution. *Sound/code/Image time itself is out of joint*, and therefore necessarily to be worked, or something to be worked by. In sum, here we see the idea of live coding questioning the very conditions by which music is made and performed – something which we can extend to other aspects of performance, such as dance.

Indeed, live coding raises the possibility that a more dynamic performance *of* code may also perhaps allow for a more dynamic engagement with other aspects of the performance, for example dance. This could be a matter of literal live coding with dance. For an interesting example of this see the work of Kate Sicchio with Alex McLean in *Sound Choreography Body Code*. [20] Here there is a 'feedback loop' between 'diagrammatic choreography' and live coding in which 'cycles of linguistic and diagrammatic construction and destruction can be observed, alongside the analogue development in music and dance' (Miriad, 2013).

Another example of dance coming together with live coding is found in the emerging culture of 'algoraves', in which people dance to algorithmic modulation. [21] The philosophy is as follows.

> These days just about all electronic music is made using software, but with artificial barriers between the people creating the software algorithms and the people making the music. Using systems built for creating algorithmic music, such as IXI Lang, overtone, puredata, Max/MSP, SuperCollider, Impromptu or Fluxus, these barriers are broken down, and musicians are able to compose and work live with their music as algorithms. (anonymous, n.d.)

However, it's not just about the live coder making music. In addition, 'It's up to the good people on the dancefloor to help the musicians make sense of this and do the real creative work in making a great party'. What algoraves make crystal clear is that an algorithm, *in action*, has a complex relationship to the abstraction of the algebraic with which digital or computational cultures are sometimes framed. Shintaro Miyazaki writes usefully here of 'algorhythm' (with a "y") (2012). He argues that computing is inevitably time bound and that an algorithm, as something that operates (we could of course say *performs*) through a non-reversible time, does so with its own rhythms. Of course, like all rhythms, these form more complex or shifting rhythms in engaging with other kinds of rhythms, most obviously in the case of algoraves those of bodies dancing. McLean writes about 'embodied programming' (2013a).

This is perhaps not a remediation or even a question of mediation so much as an immersion in the events of relation between all the aspects involved. Rather than mediation there is a kind of suspended, delayed, recursive and looped immediacy that is best named 'immediation' (Massumi, 2011: 166-167; Brunner, 2012).

Loscil—subtracting in order to convolve

I have so far discussed the way in which convolution and other uses of code in performance (and composition) open events out to a greater 'continuous variability' (Deleuze, 1993: 209). Again, one must be willing to subtract some of the major elements (or least the determinative power of these elements) that over-ride this variability. The result is something like the new electronic image as describes by Deleuze.

> The new images no longer have an outside (outside-of-field), any more than they are internalized in a whole. Rather they have [something] like a power to turn back on themselves. They are the object of a perpetual reorganization, in which a new image can arise from any point whatever of the preceding image. (1989: 265)

Taking this loss of privilege of any particular image, perpetual reorganization and a turning back of powers on themselves into questions of the performative, broadly understood, we can now understand how moves toward subtraction, or toward non-performance can lead from the old Performance to the new.

Electronica cultures in general are often committed to immanence and continuous variation at the same time as a subtraction of key performative aspects of the creation and enjoyment of music and images. Indeed, this subtraction has led to a constant questioning of the status of the "performer" (for example of the DJ or the laptop music performer) in the events involved. In such performance, we might say 'Performance' itself is subtracted, or that performance tends towards 'non-performance' (Bertelsen and Murphie, 2012; Murphie, 2011) or non-standard performance. [22] In fact, electronic/digital performance has undergone a massive transformation in tandem with an ongoing subtraction of Performance—with the 'Performer' often no longer the spectacle that they are even in other kinds of music performance (a classical orchestra or band), the 'text' no longer fixed, the engagement with a more general ecology of dynamic system of relations much more openly committed to quasi-causal interaction than control. [23]

The performance loses some or most of its centrality. It takes on various degrees of 'ambience'. As in Erik Satie's furniture music, the performance is sometimes little different to mise-en-scene. Here I will turn to that electronic music that has ambient tendencies as, first an example of subtraction in composition and performance, and second, an example of the way that convolution can find the new variability made available by subtraction.

Ambient music has many "origins". However, one of the best known stories about the origin of ambient music is Brian Eno's.

> The story of Eno being struck by a cab in 1975, lying bedridden and immobile as a record of harp music played at barely-audible levels has been recounted several times over the years. The experience of listening to music as background created for Eno, 'a new way of hearing music. And the results were immediate' (Weiner, 2004).

This is the story of an accidental subtraction, in the lowering of the volume of the music by a friend, creating an opening for a new kind of music. A new cultural technique (or at least a revival or expansion of such a technique) came into being.

One of the most interesting contemporary practitioners at the cusp of ambience—in fact usually ambient dub—today is Loscil, a project name for electronic musician (and sound designer) Scott Morgan of Vancouver. It was Loscil's discussion of his work with convolution that led to this article. Loscil takes his name from the '"looping oscillator" function in the computer music language Csound', and 'Morgan uses custom made Max/MSP sequencers and the occasional live instrument to build robust, droning soundscapes'. [24] Loscil's work's take up of convolution is exemplary in the terms I have discussed so far in this article. For one thing, it provides a wonderful example of how convolution can transform field recordings, providing a way of accessing 'continuous variability'.

The best example of Loscil's work in the terms of the discussion here is probably *Coast/Range/Arc* (2011).

> Much of the sound of *coast/range/arc* is centered around the coastal mountains of the Pacific Northwest, studded with glaciers, lakes, waterfalls, canyons and epic views. … These tracks explore the timelessness of mountainous elevations; oxygen deprived and surrounded by boundless skies. Mountains are hardly static—in fact they are dynamic on a time scale beyond

> the human experience. They grow, buckle, twist, erupt and erode at an epic pace. ... their dynamics nearly imperceptible.(Glacial Movements, 2011)

The thematic of ongoing variation in the midst of the seemingly static is reflected in the music, much of which is a combination of field recordings and convolution to transform the field recordings into ambient drone work that nevertheless still seems to carry the mood of these environments. [25] As Nick Giles writes, *Coast/Range/Arc* is 'Truly a work of stasis, and a masterwork of disciplined patience. To hell with grass: listening to this album is akin to watching mountains grow (2011). More generally, in his music Loscil focusses on reworking given sounds—of woodwind instruments for example, or the rain of his beloved British Columbia (or Cascadia as he and others call it). He uses processes such as convolution to open up the sound within a given field recording or the playing of a conventional instrument). This creates convolved sounds that he uses in the rest of the music—music that is never quite synthetic in the sense of coming from a purely technical origin (although profound synthetic in the more precise sense of being convolved).

As he says in an interview:

> I am more interested in acoustic sound sources than I am in synthesized sounds. Most of my processed sounds originate as real-world sounds that are recorded and transformed. I think this helps them maintain some sort of 'organic' quality. … I prefer a relatively inextricable sounding texture that doesn't feature any one sound over the other. (McGlaughlin, n.d.)

About another recent release, *Endless Falls* (2010) Loscil says:

> Really, they are abstract collections of blurred snapshots; foggy windows to pear into and get lost in ... The rain recordings made for *Endless Falls* were recorded in my back yard. A big part of the loscil sound makes use of convolution processing which involves taking very noisy sounds and convolving them with spectrally rich sounds to create drones and textures. I decided to use the rain sounds as my primary noisy source. (McGlaughlin, n.d.) [26]

Loscil notes that the 'initial process' of choosing field recordings 'is very important to me because it's not so much about composition as it is about getting inside the sounds themselves, processing them and shaping them into something I am engaged with' (Hampson, 2010). He uses 'processes like convolution and granular processing to alter and shape the sounds into drones and textures. Once I've got a set of sounds I like I drop them into a max/msp patch of my own making that I use to sequence and further process the sounds'.

Convolution here, even in composition, becomes a kind of performer, even a kind of 'actor'. It is not really a question of creating something entirely new, or of destroying the sound that is there. Convolution—like acting—is a technics that inhabits the edges of the virtuality of the given, in this case of sound. In *Coast/Range/Arc* the music does seem to carry something of this coast. It's not so much recognizable. It's more something like a Whiteheadian 'mood', the affective tone of the coast.

Of course, Loscil uses other methods alongside convolution. In discussing convolution it is perhaps important to note that it is a method that does not need to dominate other processes. Yet convolution, broadly conceived, has its own character. In fact, part of this is that convolution not only finds new sound within given sounds, but also activates other processes that work with the infinite differences in the relations that are already there. As a relational technics, convolution suggests a plurality of methods, not all of which are strictly speaking technological (or at least not digital). The work of another musician, Stephan Mathieu, makes this plain. For example—

> For *A Static Place* his methodology involved playing back sections of early music 78s with two mechanical gramophones, soundwaves from its period instruments being read by a cactus needle, amplified through the diaphragm and on through the horn, the ensuing sound being picked up by a pair of customized microphones before being computer-transformed by spectral analysis and convolution processes. The processes applied are based on different acoustic spaces being merged to create new imaginary spaces inhabited by the initial audio data, variously altered by transposition across media or to new listening contexts; found in translation, you might say. … What this work does in effect … is to play around with temporal trajectories, reaching back and pulling forward and beyond. *A Static Place* is his portrayal of sound's journey through time, and through several temporo-spatial dislocations. Disappeared sounds are re-captured and re-constituted through early, intermediate and current technologies. (Lockett, 2011)

One can see the same kinds of principles in other new technics of electronic/digital performance.

‘Translocal, real-time, collaborative’ performance

Work with modules and digital processing can also be collaboratively networked. Here we can consider the work of Sher Doruff on an experimental platform for networked performance. Doruff, who with others developed the experimental networked, multi-media software *KeyWorx* (previously named *KeyStroke*) draws attention both to the complexity of collaborative negotiation and the extension of theatrical performance in the use of computing in networked performance. [27]

> We wanted to have the same kind of modularity in an interface but much more intuitively rendered. People would need to enter and somehow easily understand what was happening in a given patch.
>
> But also, and this was unique to *KeyStroke*, parameters of every media object and controller would be transparently open so that, for example, I could change the speed and position parameters of one of your movie clips with the frequency and amplitude of one my sound clips. That sort of activity opened across a full range of devices and objects. So basically you needed to adapt to creative negotiation on every level. It’s collaborative like a theatre piece is collaborative yet more so. … All media and controllers (mouse, camera, joystick, GPS, etc.) were objects in an open field. (in Doruff and Murphie, 2012)

KeyWorx was an important platform for some of the earlier networked performances. In some ways it remains ahead of what came after. For one thing, the fact that each person’s interface and configuration could be changed by everyone else’s interface made for an intense experience of self and other, of location and network, perhaps of location in the network. Doruff calls the performance involved ‘translocal, real-time collaborative performance’.

> … translocal experience, as an embodied experience, is amplified because your body is … you have the sensation of … how can I say this? The effect of intensities of translocal performance when it’s indeterminate and you’re collectively negotiating and making choices together and you’re playing off those choices as a jazz band would or as a

> dancers might … the translocal becomes incredibly, intensely physicalised, but your feeling of space is non-locally oriented. (Doruff in Doruff and Murphie, 2012)

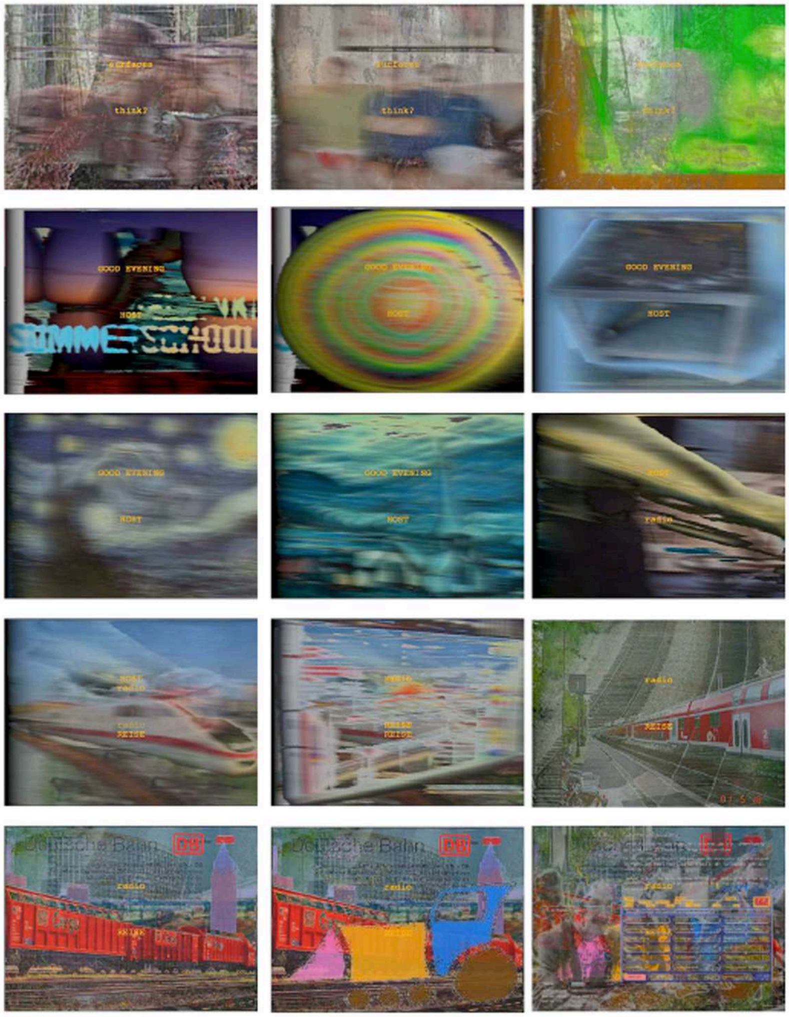


Figure 7—KeyWorx screen shot sequence from the performance by Lodewijk Loos in Rotterdam and Eric Redlinge in New York. Taken, with permission, from Connected! LiveArt (Doruff, 2005: 38)

I would suggest that *KeyWorx* provided a general "convolutionary" performance space.

> You really find yourself intensely in this differential space. … What I'm interested in is collaborative performance as transducer through non-local space. The space becomes … it's not that it's a non-issue, it's just a completely different issue. (Doruff in Doruff and Murphie, 2012)

As Doruff well understands, in the signal complexity that results, performance itself is often a matter of layers of emergent events and structured negotiations.

Ongoing Convolution as Cultural Condition

I have more than hinted throughout that convolution is not only a technical or mathematical process. We have seen that convolution is an open way of diagramming, or rediagramming, folding the world, in situ. Ongoing convolution is *an ongoing cultural symptomatic or condition, a part of how culture works,* as well as a set of technics enabling a variety of technologies and techniques. A culture engaged with computational/signal processes like convolution (or live coding, or networked performance) is a culture attuned to different aspects of the world. To a large extent, especially in such a culture, *we are our technics*. Technics and technologies are the way we modulate our becoming.

I am not however, suggesting that convolution, or even computational processes are *the* cultural condition of our times. Or if they were, it would be precisely because they allow for the complex mixing of conditions. Computational and signal processes are not so much foundational as they are communicative with other aspects of culture. As technics, convolution and other computational processes open out to many other possible aspects of process. This requires us to rethink significant aspects of process itself.

Endnotes

1. See also http://www.loscil.ca/.
2. Others who have worked on this question include Mackenzie (2011), Parisi and Portanova (2011), Munster (2010; 2011), McLean (2010; 2013), Miyazaki (2012) and Manning (2006).
3. I am grateful to the Department of Aesthetics and Communication at Aarhus University for hosting and engaging with my research into these issues in 2011. I am also grateful to the Australian Research Council for the Discovery grant that funded the development of these ideas.
4. This article was in part motivated by Bodil Marie Stavning Thomsen's innovative work on the move from sign to signal in thinking about media and art. See Thomsen, 2012 and Thomsen, Sundholm and Jørgensen, 2012.
5. See Auslander, 2008: 2ff for an extended discussion of 'liveness' and its ambiguities. On technology, media and liveness, see also Murphie, 1990, Scheer and Klich, 2011, and Cull, 2012.
6. This is meant in the sense in which Simondon discussed information, in which form is always informed and information is always as much a material question of formation as anything else. See Shaviro, 2006.
7. See http://chunkymove.com.au/Our-Works/Current-Productions/Mortal-Engine.aspx
8. See Munster, 2010 and 2011 for a discussion of embodiment, signal, cross-signal processing, generative differentials, synthesis and synaesthesia. On 'differential media', and on the relation between electronic music and a theory of 'differential media', see Murphie, 2003 and 2004.
9. On Fourier and heat, see the excellent BBC documentary, *A Short History of Mathematics*, episode 3. For a much more precise summary of Fourier's work on sound, see Evens, 2005: 3ff.
10. More complex sounds consist of a main frequency overlaid with a series of other frequencies, or harmonics. It is this that makes for the different sounds of a violin or a piano. Most sounds are of course a complex mix of frequencies, amplitudes and so on, varied over time. It perhaps helps to think of the complexity of waves on the sea.
11. For more on Fourier with regard to the implications for theorising coding see Mackenzie, 2011: 73-74.
12. This is Whitehead's description of the false division of nature into the mind/perception on the one hand, and the object perceived on the other (Whitehead, 2009: 26). Here however I am broadening the notion of bifurcation. Convolution is literally an "un-bifurcator".

13. Norbert Wiener discusses convolution and it is important to Shannon and Weaver's famous mathematical model of communication. The best discussions of convolution I have found are Roads, 2004 and Wishart, 1994. I am particularly grateful to Mat Wall-Smith for first discussing convolution with me, in particular for his observation that the implication of convolution is that any existing sound allows infinite permutations and explorations, with convolution a particular flexible technical procedure via which this could happen. Many thanks also to Morten Breinbjerg for his patient explanation and help on this issue, and to Kevin Wong for explaining convolution's relation to time.
14. This impulse response results from as short a sound as possible (preferably and impossibly the duration of the sound sample should be zero), with a full range of frequencies, such as a gun shot.
15. Mackenzie, 2011 discusses wi-fi, convolution and deconvolution extensively.
16. This is image is not, strictly speaking, live coding. It nevertheless exemplifies the way that programming languages can generate images. In this case the programming language is Piet (http://www.dangermouse.net/esoteric/piet.html).
17. Fluxus is a '3D game engine for livecoding worlds into existence'. See McLean, 2010 and http://www.pawfal.org/fluxus/
18. See also http://toplap.org/. Alex McLean's web site (http://yaxu.org/) and articles provide a very good way into understanding live coding. McLean is both a live coder and a leading scholar in the area.
19. Slub is a music group using generative software that includes Alex McLean, Adrian Ward and Dave Griffiths. See http://slub.org.
20. See the video 'Sound Choreography Body Code', http://vimeo.com/62323808 and see also Sicchio's site at http://www.sicchio.com/ for her work on dance and media.
21. On algoraves, see http://algorave.com/about/ or see the video here http://tobiasreber.tumblr.com/post/48598679032/music-and-live-coding-as-activity-guest-post-by-alex (McLean, 2013b).
22. In parallel to, and different to, something like Laruelle's 'non-philosophy' or non-standard philosophy. Bertelsen and I draw the notion from Deleuze.
23. Quasi-causality is 'co-causal, requiring the contribution of objective conditions ... combined just so' (Massumi, 2011: 190).
24. http://ghostly.com/artists/loscil. See also http://www.loscil.ca
25. The track Fromme can be heard at http://www.youtube.com/watch?v=It3MztrGAjs
26. The track Endless Falls can currently be heard at http://www.youtube.com/watch?v=Kyez8QxweSc
27. http://www.keyworx.org

References

Anonymous. 'About' algorave.com (n.d.) http://algorave.com/about/

Auslander, Philip. Liveness: Performance in a Mediatized Culture (London, Routledge: 2008).

BBC. A Brief History of Mathematics (podcast series) (n.d.), http://www.bbc.co.uk/podcasts/series/maths

Bertelsen, Lone and Murphie, Andrew. 'Affect, Subtraction and Non-Performance', Peripeti: tidsskrift for dramaturgiske studier, 17 (2012): 78-87.

Breinbjerg, Morten. (200) 'Poesis of Human-Computing Interaction: Music, Materiality and Live Coding', in Christian Ulrik Andersen and Søren Bro Pold (eds.) Interface Criticism: Aesthetics Beyond Buttons (Aarhus, Aarhus University Press, 2011), 162-177.

Brunner, Christoph. 'Immediation as process and practice of signaletic mattering', Journal of Aesthetics & Culture 4 (2012), http://dx.doi.org/10.3402/jac.v4i0.18154

Cox, Geoff and McLean, Alex. Speaking Code: Coding as Aesthetic and Political Expression (Cambridge, MA: MIT, 2013).

Cull, Laura. Theatres of Immanence: Deleuze and the Ethics of Performance (London: Palgrave Macmillan: 2012).

Deleuze, Gilles. Cinema 2: The Time-Image (London: Athlone, 1989).

Deleuze, Gilles. 'One Manifesto Less' in Constantin Boundas (ed.) The Deleuze Reader (New York: Columbia University Press, 1993): 204-222.

Doruff, Sher. (ed.) Connected! LiveArt (Waag Society, Amsterdam: 2005), http://spresearch.waag.org/images/connected-print.pdf

Doruff, Sher and Murphie, Andrew 'Multimedia Mixing and Real-time Collaboration: Interview with Sher Doruff about the development and use of *KeyWorx*, the Translocal and Polyrhythmic Diagrams' Fibreculture Journal 21 (2012), http://twentyone.fibreculturejournal.org/fcj-153-multimedia-mixing-and-real-time-collaboration-interview-with-sher-doruff-about-the-development-and-use-of-keyworx-the-translocal-and-polyrhythmic-diagrams/

Evens, Aden. Sound Ideas (Minneapolis: University of Minnesota Press, 2005).

Giles, Nick. 'Loscil – Coast/ Range/ Arc (Glacial Movements)', Cyclic Defrost (2011), http://www.cyclicdefrost.com/blog/2011/09/loscil-coast-range-arc-glacial-movements/

Glacial Movements. 'GM010 LOSCIL "coast/range/arc"' (2011), http://www.glacialmovements.com/gmcat4.htm

Hampson, Simon. 'Symbiosis Episode 45—Loscil', Symbiosis (2010), http://symbiosis.com.au/2010/06/mixes/symbiosis-episode-45-loscil/

Lockett, Alan. 'A Static Place: Stephan Mathieu', Furthernoise (October, 2011), http://www.furthernoise.org/page.php?ID=408.

Mackenzie , Adrian. Wirelessness (Cambridge, MA: MIT Press, 2011).

Manning, Erin. 'Prosthetics Making Sense: Dancing the Technogenetic Body', the Fibreculture Journal, 9 (2006), http://nine.fibreculturejournal.org/fcj-055-prosthetics-making-sense-dancing-the-technogenetic-body/

Manning, Erin. Relationscapes: Movement, Art, Philosophy (Cambridge, MA:MIT Press, 2009).

Massumi, Brian. ‘The Thinking-Feeling of What Happens’, Inflexions 1(1) (May, 2008), http://www.inflexions.org

Massumi, Brian. Semblance an Event (Cambridge, MA: MIT Press, 2011).

McGlaughlin, Liam. ‘Loscil Interview with Scott Morgan’, Soundshock (n.d.), http://soundshock.co.uk/index.php/feature-band-interviews/1598-loscil-interview-at-soundshock

McLean, Alex. ‘Visualisation of Live Code’, Alex McLean (May 29, 2010), http://yaxu.org/visualisation-of-live-code/

McLean, Alex. ‘What is Embodied Coding?’, Yaxu.org (2013a), http://yaxu.org/what-is-embodied-programming/

McLean, Alex. ‘Music (and live coding) as activity’, a100ql: a hundred quirky legs (April 22, 2013b), http://tobiasreber.tumblr.com/post/48598679032/music-and-live-coding-as-activity-guest-post-by-alex

Miriad. ‘Audio:Visual:Motion [REDUX] - Sicchio and McLean - Sound Choreography Body Code’, Miriadonlie (2013), http://vimeo.com/62323808

Miyazaki, Shintaro. ‘Algorhythmics: Understanding Micro-Temporarlity in Computational Cultures’, Computational Culture 2 (2012), http://computationalculture.net/article/algorhythmics-understanding-micro-temporality-in-computational-cultures

Munster, Anna. ‘Syn-aesthetics–Total Artwork or Difference Engine?‘, *Inflexions* 4 (2010), http://www.inflexions.org/n4_munsterhtml.html

Munster, Anna. Materializing New Media: embodiment in information aesthetics (Dartmouth College Press: Hanover, New Hampshire, 2011).

Murphie, Andrew. 'Negotiating Presence: Performance and New Technologies' in Phillip Hawyard (ed.) Culture, Technology & Creativity: In the Late Twentieth Century (London: John Libbey and Co., 1990): 209-226.

Murphie, Andrew. 'Electronicas: Differential Media and Proliferating, Transient Worlds', fineart forum 17:8 (August, 2003).

Murphie, Andrew. 'The World as Clock: The Network Society and Experimental Ecologies', Topia (Spring, 2004):117-139.

Murphie, Andrew. 'Performance as the Distribution of Life: from Aeschylus to Chekhov to VJing via Deleuze and Guattari' in Laura Cull (ed.) Deleuze and Performance (Edinburgh: University of Edinburgh Press, 2009): 221-239.

Murphie, Andrew. ‘On Not Performing: the Third Enclosure and Fractal Neofeudal Fantasies’, *Scan* 8:1 (2011), <http://scan.net.au/scan/journal/display.php?journal_id=156>

Murphie, Andrew. ‘Hacking the aesthetic: David Haines and Joyce Hinterding’s new ecologies of signal’, Journal of Aesthetics and Culture 4 (2012), http://dx.doi.org/10.3402/jac.v4i0.18153

Parisi, Luciana and Portanova, Stamatia. ‘Soft Thought (in Architecture and Choreography)’, Computational Culture, 1 (2011), http://computationalculture.net/article/soft-thought

Roads, Curtis. Microsound (Cambridge, MA: MIT Press, 2004).

Scheer, Edward and Klich, Rosemary. Multimedia Performance (London, Palgrave Macmillan: 2011).

Shaviro, Steven. ‘Simondon on Individuation’, The Pinocchio Theory blog (January 16, 2006), http://www.shaviro.com/Blog/?p=471

Smith, Steven W.. The scientist and engineer's guide to digital signal processing (San Diego: California Technical Publication, 1997).

Stiegler, Bernard. ‘Desire and Knowledge: The Dead Seize the Living’, Ars Industrialis: association internationale pour une politique industrielle des technologies de l'esprit (n.d.), http://www.arsindustrialis.org/desire-and-knowledge-dead-seize-living

Thomsen, Bodil Marie Stavning. ‘Signaletic, haptic and real-time material’, Journal of Aesthetics & Culture 4 (2012), http://dx.doi.org/10.3402/jac.v4i0.18148

Thomsen, Bodil Marie Stavning, Sundholm, John and Jørgensen, Ulla Angkjær. ‘Foreward’, Journal of Aesthetics & Culture 4 (2012), http://dx.doi.org/10.3402/jac.v4i0.188192012

Warburton, Dan. ‘Radioland’, The Wire 295 (September, 2008): 53.

Weiner, Matthew. ‘Brian Eno and the Ambient Series, 1978-1982’, Stylus Magazine (2004), http://www.stylusmagazine.com/articles/weekly_article/brian-eno-and-the-ambient-series.htm

Whitehead, Alfred North. The Concept of Nature (Ithaca: Cornell University, 2009).

Wishart, Trevor. Audible Design (London: Orpheus The Pantomime Ltd, 1994).

Wolfram Mathworld. ‘Convolution’, (n.d.), http://mathworld.wolfram.com/Convolution.html

Zhen, Waigai. Linear Networks and Systems: Fourier analysis and state equations, Volume 1 (New Jersey: World Scientific Publishing, 1990).