11

Computational Aesthetics

M. Beatrice Fazi and Matthew Fuller

It is the contention of this chapter that computation has a profound effect on the composition of digital art. We understand computation as a method and a force of organization, quantification, and rationalization of reality by logico-mathematical means. The computational precedes yet grounds the digital in its technical, social, and cultural manifestations: it finds in digital technologies a fast, efficient, and reliable technique of automation and distribution, yet remains a notion wider and more powerful than the digital tools that it subtends. Art, operating with the digital prefix and taking on many of the characteristics of the contemporary world, is inherently interwoven with the specific features of computational structures. At the same time, though, it can be said that aspects of digital art have yet to be sufficiently considered from that perspective. To some extent this is understandable, given the immense flexibility—and, often, resultant opacity—of computational systems. Digital art, however, builds upon and works through the computational, sharing its limits and potentials while also inheriting conceptual histories and contexts of practice. For this reason, we contend that an aesthetics of digital art is, at a fundamental level, a computational aesthetics.

Medium Specificity

The crux of our argument can be summarized in the particular kind of medium specificity of the aesthetics of digital art, a specificity that we see pertaining to this art's primary computational character. When making a claim for the computational specificity of digital art, however, we abstain from flattening this proposition onto openly "modernist" arguments, or following on with the sets of uneasy qualifications and rejoinders that come after such positions. We are wary of the essentialism that such an argument would imply, mourn, or efface. It is our contention, however, that the risk of "computational essentialism" is diminished by the nature of computation itself. It is somewhat perverse to look to Greenberg as a point of orientation, but it

A Companion to Digital Art, First Edition. Edited by Christiane Paul.

will serve to make the point: traditionally, a modernist medium-specific aesthetics would call for the individuation of a "raw" materiality, which—in operation and in effect—amasses and defines the potential for artistic expressivity of a certain medium, a modernism of attenuation (Greenberg 1961). In the case of computational aesthetics, however, such a prescription is more difficult to sustain. How is one to match the material promises of a medium if this medium does not have an idiotypic substantial form (such as canvas, paint, marble, or mud), but rather has to be understood as a method and a force that, through rules, constraints, and capacities for expression, continually renegotiates its own structures of existence? In other words, what makes computation special in terms of its mediality—and thus perhaps different from any other media coming into composition with art—is the impossibility of describing it simply as an intermediary "substance."

Art, thankfully enough, is not simply communications. The relation of art with its media has been complex— a relation that is disavowed as much as it is explored, and through which one can trace the histories of many modalities or kinds of art that may themselves not cohere into a stable lineage. Art always propagates, rather than necessarily progresses, by disrupting and reinventing its terms of growth and domains of operation. Computation, however, has, in a certain sense, been a more historically delimited domain. To some extent this is due to the relatively young state of the field as an organized discipline. At the same time, we argue, computation's development through mathematics, logic, philosophy, and physical engineering gives it an equally rich genealogy. With its folding out into culture and the social, and indeed in its entanglement with art, it is undergoing further mutation, and its complex lines of invention and imagination find new forms of growth.

Recognizing this, critical discourse in recent years has developed cultural and artistic understandings of some of the mechanisms and components (algorithms, values, parameters, functions, codes, and so on) through which computation operates, for instance via the emergence of fields such as software studies (Fuller 2008). We would like to supplement this discussion with a consideration of computation's mediality as a mechanism of ontological and epistemological production. In terms of our medium specificity argument, this implies that computation is a medium in so far as it actualizes modes of being, levels and kinds of agency, and procedures of thought and configuration. The ontological and epistemological expressions of computation are concretized and become operative at various scales: in the cultural, the societal and the political, as well as in art and elsewhere. Through a double articulation, computation changes these fields yet maintains its own specificity; a specificity that is in turn affected, in variable ways, by the mutational forces of these fields' characteristics. Calling for a recognition of the medium specificity of computation in digital art thus means to take up the challenge of considering a mediality that surpasses the bounds of its grossly material instantiations and circumstances. In fact, acknowledging medium specificity involves reconsidering the notion of matter altogether, via the mobilization of all categories of the computational (whether sensuous, or logical, or both), and in light of the ontologies and epistemologies that computational systems initiate or participate in.

A problem that immediately follows from this argument about medium specificity is how computation can be understood and spoken of, and by which means its consequences in the area of digital art can be mapped out. In attempting to address these questions, we do not advocate a "programmatic aesthetics," but a way of understanding the things that are explicitly or implicitly taken into account when working with computational systems. Computational aesthetics is certainly partially entwined with the computing machine, and in many particular works founded on very specific articulations of that interlacing. Yet the existence of computational aesthetics is not exclusively tied to a particular past, present, or future technology. Computation, we contend, is a systematization of reality via discrete means such as numbers, digits, models, procedures, measures, representations and highly condensed formalizations of relations between such things. To compute involves abstractive operations of quantification and of simulation, as well as the organization of abstract objects and procedures into expressions that can (but also may not) be thought of, perceived, and carried out. Attending to computational aesthetics, then, puts in question the forces of all degrees and kinds that participate in these abstractions, and enquires what level of autonomy one should assign to such forces and abstractions. Similarly, a medium-specific computational aesthetics addresses the ways in which other techniques and genealogies (e.g., language, science, mathematics, and art itself) conjoin, contribute to or contrast with computation, and thus result in often irreconcilable, convulsive or, conversely, reductive interrelations of other aesthetic approaches, ontological commitments, knowledge structures, and arenas of practice. The impact of computation on other hitherto distinct fields constitutes, to a large extent, the status of the problematic of contemporary forms of life. We can therefore conclude that computation is as much a condition as it is a medium.

Computational Construction

It is in light of these and related issues that the condition of computational aesthetics has to be understood not as given but constructed. This claim, however, comes with two important qualifications.

To construct is to build up, to compose, to compile. A construction requires, in varying measures, a dose of planning and an amount of improvisation, the laying of foundations and the addition of decoration, the work of an engineer and the effort of a craftsperson. In this sense, a construction is less the straightforward manufacture of a result or an output than a heterogeneous process of creation. Constructing a computational aesthetics is a similarly inventive and procedural endeavor. It is, we claim—alongside the recognition of ecology and the invention of economies—a requisite for contemporary thought, imposing key issues proper to 21st-century culture. For example, questions such as how to define numerically determined rules for the analysis, codification, and prediction of the world; how to account for digitally interfaced modes of sensing; and how to theorize new spatio-temporally distributed and networked prospects for cognition.

If it is a truism that computational technologies have brought about a fundamental epistemological break,[1] constructing a computational aesthetics means to come to terms with both the disruptions and the opportunities that this break initiates in modes of perceiving, acting, and cognizing. In fact, it involves coming to terms with these conditions while looking for and articulating computational aesthetics' internal epistemological validations—those that are inherent to the theories and practices of computation itself. The construction of computational aesthetics, therefore, calls for a reworking of many of the conceptual categories, classes, types, and criteria involved in aesthetics, noting that aesthetics is in turn understood here as a theory of construction—again!—of what constitutes experience. In other words, we are arguing, on the one hand, that to understand digital art in terms of an aesthetics of computation is key to the status of contemporary culture, which indeed is a computational culture. On the other hand, however, the very notion of computational aesthetics for us goes well beyond a theory of "art made with computers," and becomes an investigation of the more foundational and formative aspects of the reality of the computational itself. In this respect, the reworkings of the aesthetic that we are here advocating are acts of both discovering and inventing the unfamiliar, the nameless, that which has been forgotten and is yet to be known: computational aesthetics must construct its own concepts.

Our first qualification of computational aesthetics' mode of construction should be read in light of what we consider the restrictions or limitations of a traditional "constructivist epistemology" for addressing the potential for conceptual discovery and invention. To claim that computational aesthetics is not given, but that it has to be constructed, would seem to echo the slogans of social constructivism, according to which situations are constructed by the interpretations that humans give of them. While there are some conditions and circumstances in which such an approach may gain significant traction also in digital art, we are keen to stress that the sociocultural constructivist position is not what we argue for, and that the construction of computational aesthetics advocated here is irreducible to the social constructivist epistemological paradigm. We would like to take a distance from the sociocultural constructivist agenda to extend the significance of "construction" from an epistemological level to an ontological one. Which is to say: when constructing computational aesthetics one creates not only ways of knowing reality, but reality itself. To be more explicit, we understand the construction of computational aesthetics as a process that is "internal" to the notion of computation, and should therefore not to be approached from any particular disciplinary ground. Computer science alone cannot fully account for the modes of existence of the aesthetics of computation, but neither can cultural theory, philosophy, or art. To say that computational aesthetics is not inferred from some particular disciplinary area, however, also means that its actuality cannot be subsumed under individual categories such as the societal, the cultural, and the economic, or of course the aesthetic, although this actuality can surely be more or less successfully interrogated from such perspectives.

Computational aesthetics does not arise from a void; it is of course part of society, culture, and economy—if we can, for a moment, accept the ruse that these things are adequately nameable. At the core of this issue lies, for us, the following point: to understand construction as the methodology proper for an immanent investigation of computation. We believe that social and cultural constructivism, in wanting to accommodate and assimilate difference, reiterates instead a "transcendent" take on computational practices and technologies. From this transcendent perspective, human social histories or human cognitive processes are equally relative amongst each other, yet still causally superior to the events that they are said to construct. We argue for another view: that the construction of computational aesthetics is not solely based upon the determinism of a particular identity-forging coordinate, such as a time in history, or a group of people, but that this construction is in fact incidental to computation's capacity of being an immanent operation of production of its own as well as other entities' modes of existence. Computational aesthetics is not produced by the social but is social. Similarly, it is not the result of a certain culture; it is culture. The diversities of the planes into which computational aesthetics cuts are not the transcendent cause of the aesthetics; these planes and multiplicities of contexts, intentions, norms, actions, perceptions, etc. must themselves—to appropriate Deleuze and Guattari's claim—be made (Deleuze and Guattari 2004, 7). With this assertion we do not mean to say that in the aesthetic investigation of computational media anything is equal to anything else. On the contrary, we affirm that the realities of computational aesthetics are produced in the expressions that the aesthetics of computation finds for itself. The construction of such aesthetics is always in the computational event.

Having clarified this, we should note that, while we are wary of a simply sociocultural constructivist approach, our position also differs from what one could call an "autopoietic" constructivism that would frame construction as a self-producing and self-organizing operation of subjective experiencing. This then is our second qualification of computational aesthetics' construction—a qualification that perhaps can help us to clarify our proposal for an immanent investigation of computational aesthetics. According to the autopoietic dialectics between observing and observed systems, everything relates to the environment in so far as it establishes and stabilizes itself in relation to it. The observer thus constructs her own world self-reflexively—that is, by positioning herself in relation to an environmental situation (Maturana and Varela 1992). Without disavowing the importance of autopoietic constructivism for some fields (such as theories of cognition, which see it as variously involved with the world), we believe that this type of constructivism becomes particularly problematic when applied to computational aesthetics. In our opinion, autopoietic approaches to digital art seem to overlook the fact that computation is full of encounters between levels of expressivity and actuality that cannot interact in terms of subjects and objects, or within the confines of an environmental "outside" or an "inside" of the system.[2] Many of these encounters or determinations in fact concern the (human) users of computation, not computation itself. We believe instead that the construction of computational aesthetics also involves incongruences and incompatibilities: in computation there are many particular cases but there is also an at least implied pretense to universality; the different speeds of eternity and fracture are often disjointed, and the diverse scales of what is too big to count or too small to see are frequently beyond subjective perception. In this sense, the construction of computational aesthetics needs to be radicalized from within the limits and potentialities of the computational itself, and not imposed upon the experiential positioning of an observer (for whoever or whatever this latter is supposed to be). In other words, what we are advocating here is the capacity of computational aesthetics to not simply represent reality, but to contribute to the immanent constitution of reality itself.

Ten Aspects of Computational Aesthetics

In order to cut into the condition of computational aesthetics, we would like to offer a short overview of some of the features and characteristics that, to a greater or lesser extent, and to varying degrees of combination, articulate the reality of computation. It should be stressed that we are not looking for the ultimate qualities and values of either computation or aesthetics. Rather, we take these characteristics and features as modes of existence of the computational that infiltrate (and, in some cases, pervade and direct) its ontological and epistemological productions. In other words, these features and characteristics inform computation's modalities of being, levels of agency, and procedures of thought that mark the medium specificity of digital art, on the one hand, and its constructive nature, on the other. There is therefore no claim that the list below is either an exhaustive itemization of the conditions of computational aesthetics, or that aspects or combinations of it are exclusive to computational aesthetics and do not surface in other contexts and kinds of art, or aesthetics more broadly. What we suggest, instead, is that computational aesthetics brings the modes below into a sharper focus and degree of compositional strength. If aesthetics can be understood as a theory of how experience is constructed, then this list attempts to account for some of the modalities of the computational that partake in such constructions. In some cases the items on this list help to sustain those constructions and to bring them into the empirical realm; in others they clash with the very category of experience altogether. The examples we offer are equally meant to provide an illustration of how computational aesthetics produces, regulates, but also points beyond its own ontological and epistemological validations, and thus always has to be found and investigated in the computational event.

1. Abstraction and Concreteness

Computation sets in motion some fundamental reorientations of culture, and of the circumstances in which art occurs, in that it endures as a conjoint condition of the abstract and the concrete.[3]

On the one hand, computation is a technique of abstraction. Layers of abstractions are piled up, from the hardware and the machine language right up to the graphic user interface; they manage the in-betweens of electronic circuits and symbolic procedures, and thus safeguard the operability of computing machines. In this respect abstraction is a self-contained dimension of existence of the computational. Historically and conceptually, computation draws upon the formal abstractions of logic and mathematics. Abstract mechanisms of inference drive it, while formal languages and symbol manipulation are among the abstract means that ground the very possibility of algorithmic "effective procedures."

On the other hand, however, computation is as concrete as the world in which it participates. Computation not only abstracts from the world in order to model and represent it; through such abstractions, it also partakes in it. In this sense, computation is a technology of material agency: there are the actions of algorithms organizing commercial warehouses, air traffic, and administrative records; there are the social associations of networked practices, which aggregate and shape both the technological and the cultural; there are the solid effects of software applications which intervene in and bring about modes of knowing, trading, writing, playing, perceiving, interacting, governing, and communicating.

The qualities of abstraction and concreteness have innumerable effects in terms of the constructivism and medium specificity of computational aesthetics. One of those is that the abstract structures of computation can move fast from one instantiation or occurrence to another (an algorithm can be used for sorting rice grains, faces, or patterns of pixelation, for instance). The movement across multiple sites and occasions of a work is one way of tracing the variable characteristics of computational aesthetics across social forms, and to highlight some of the ways in which the computational is often built into the latter. Tracing aspects of such a genealogy, the work of YoHa (Matsuko Yokokoji and Graham Harwood) and Graham Harwood—epitomized in *Lungs (Slave Labour)* (2005, with Mongrel);[4] *London.pl* (2004);[5] and *Coal-fired Computers* (2010, with Jean Demars)[6]—amongst others, works with the abstractions of relational database systems to both concretize their schema and establish relational calculus as a grammar of composition that links labor, primary accumulation, mechanisms of power, and the materialities of organic and non-organic forms of life.

2. *Universality*

Universality in computation is established at the conceptual level of the machine. In the 1930s the computer science pioneer Alan Turing famously developed a thought experiment that put the mental activity of computing in mechanical and finite terms in order to imagine a universal device (subsequently known as the Universal Turing Machine) that would be able to replicate the behavior of any other Turing machine. Anything mechanically computable could and would be computed by such a universal device, as it would be capable of processing any algorithm fed into it (Turing 1936). The Universal Turing Machine (UTM) is the basis of the "von Neumann architecture" for stored-program computers (Davis 2000), and thus the foundation of present-day computing devices, which are in fact general-purpose and can be programmed to emulate each other. Such functional generality is perhaps the most basic, yet crucial conceptual premise of modern computational endeavors. Moreover, the amplification of functional operations also underpins the possibility of understanding computation as a technique of abstraction that is already geared toward universality. Computation generates and disseminates abstractions that are general and inclusive. In its renegotiation of structures of existence, computation aims to encompass and produce the universality of formal methods of systematization and of all-purpose models of reasoning.

Artists may respond to this universality by attending to the specificity of particular instantiations of computational forms, as in the wave of attention that has been paid to retro-computing platforms and to Game Boy hacks. The artist group Beige (Cory Arcangel, Joe Beuckman, Joe Bonn, Paul B. Davis), for example, looked for constraints in order to address the question of universality and thereby reveal universality's nature by noting its particular, historical concrescence in styles of design, the development of genres of game, and so on.[7] In their work, computing was always filtered through the quirks and constraints of a particular system with all its clunkiness and idiosyncracy. *Super Abstract Brothers* (2000), for instance, replaced the sprites and landscape of a Nintendo Entertainment System cartridge game with blocks of color.[8] Alternatively, the nature of claiming universality itself may be a foundational point of exploration for certain artists, such as David Rokeby,[9] whose multifarious and rigorous works probe and ally with the various ways in which algorithms and other procedural and interpretative acts of computers shape and condense as spaces and behaviors. One example amongst many would be *n-Cha(n)t* (2001), where a circle of computers run generic speech-recognition and speech-to-text programs. The machines hear and interpret each other, responding with further speech. The resulting cycling mixture of interpretation and response also makes the implicit offer for people to loop themselves into the feedback cycles of chuntering vocalizations and tangential interpretations in which machine intuition, logic, and chance are intriguingly interwoven.[10]

3. *Discreteness*

Something is defined as discrete if it is disjointed, separated, distinct, detached, or discontinuous. Discreteness arguably is the hallmark of the digital. Digital systems by definition are discrete systems: they represent, store, and transfer information in terms of discontinuous elements or values (binary digits, for example). The computational, as we explained above, is not synonymous with the digital: the digital should be understood as an automation of the computational. Computation itself, however, is also marked by discreteness. As a rule-governed activity, computation arranges calculation procedures through the sequential succession of countable and separable states. The "valid reasoning" that computational mechanisms are meant to encapsulate and model is explicated via the manipulation of quantifiable entities into a finite sequence of well-defined steps. The time and memory that such a sequence employs in order to perform the computation are also finite; so too are the input that set it in motion and the outcome that it generates.

The discreteness of computation is often at odds with the continuity of interactions proposed by affective philosophies, system theories, and cognitive phenomenologies, which—in art, culture, and science—focus on the dynamic becoming of relations and connections. Discreteness also prompts questions about the recognition one is willing to give to computational entities, on the one side, and computational processes, on the other, and invites further investigation whether a theoretical and technical reconciliation between the two is possible.

The discreteness of a computational object may also be used for comedic effect, as in the game *Surgeon Simulator* (Bossa Studios 2013) where a patient, body parts, surgical tools, transplant organs, and the paraphernalia of an operating theater are all to be worked on by interacting with a First Person Shooter (FPS)-style interface. The game is a device-based Grand Guignol that reaches the level of slapstick in the clashing, clumsy, interactions of the handled objects' levels of discreteness.

Discreteness also allows for the recomposition of things and their fixture in new positions, thereby generating both new kinds of commodity forms, and new commonalities. This potential is illustrated in its violation by Yuri Pattison's *e ink pearl memory* (2012), in which discreteness is employed as a means of disrupting the transcendental role of the commodity. Here, amongst other forms and arrangements of informational matter, such as a small spill of photocopier toner, treated Kindle e-book readers are modified to display fixed abstract images, that is to say they are 'broken', becoming finite, discrete. Conversely, the discreteness of digital material such as software or music also allows for an establishment of its sharing or commonality in certain ways. This aim is a prominent aspect of numerous computational initiatives, but becomes most obvious in Free Software, and in file-sharing ventures such as the Pirate Bay, a project itself sometimes manifesting explicitly as artways through the Piratbyrån (Bureau of Piracy).[11] As different levels of discreteness—alongside the ability to copy wholes or parts—combine with computing resources such as memory or bandwidth, discreteness also plays a part in the development of expressive styles and modes of computing at multiple scales, including the determination of degrees of granularity in the resolution of images.

Discretion, the ability to keep things separate, is by necessity a key factor of the ethico-aesthetic dimensions of computational technologies' political nature,[12] determining what they reveal or make tractable, as well as what they hide. The regimes of what links, what can be analyzed, what pools together, and what remains apart are core to the nature of such systems.

4. *Axiomatics*

Computation is axiomatic. The modes of abstraction, universality, and discreteness discussed above are key features of the axiomatic character of computational systems, and of the many parallels between computation and logico-mathematical reasoning. Via this axiomatic character, these parallels distinguish past and present conceptions of digital computing machines.

As is well known, Alan Turing (1936) established the notion of computability after discovering a particular class of numbers that cannot be computed. Turing demonstrated, however, that what is computable can be determined algorithmically; that is, through a mechanized principle of deductive inference, "with every tiny step of reasoning in place" (Chaitin 2005, 30). What had been an informal notion of computation was hence formalized into the axiomatic parameters of the logico-mathematical comprehension of correct inference. In this sense, to compute became to manage discrete quantities, and to do so by following abstract and finite inferential rules with universal applicability.

Today the axiomatic nature of computing subsists in and thrives on its many formalisms. The axiomatic method is central to symbolic notation and procedural execution: for every calculating process, whether a basic operation or a more convoluted function, the computational system engages, and then reiteratively re-engages again, with the general problem of determining consequences from a handful of validly symbolized premises. It is because of the unavoidability of axiomatics in computation that digital art and theory alike cannot leave the formalizations of computation to the computer scientist's classroom. Instead, they need to take up the challenge of thinking and creating an aesthetics of computation that takes into account, if not limiting itself to, the inferential and rule-based character of computational systems, while remaining aware of the ways in which computation borrows methods from mathematics and logics.

5. *Numbers*

Computation holds a multifaceted and profound relationship to numbers. Of course, contemporary computers are "metamedia" (Manovich 2013) capable of accomplishing much more than merely "crunching" numbers. However, the computing machine's relation to the numerical remains intimate. This is due partly to computation's discrete and quantitative nature, and partly to the fact that a computing machine has to operate within the parameters of a calculation.

The very idea of number has continuously changed over time, stretching and convoluting to encompass new categories and attributes, and has become something different again in its encounter with the medium specificity of computation: a means of establishing relations among abstractive methods, formal systems, and concrete tasks that are governed in turn by the operation of numbers. Although this recursion to some degree existed before in techniques such as the calculus, it is fundamentally different in computation in terms of the quantity and density of operations. Numbers in computation show various qualities and behaviors. As a unit of measurement, numbers are used, for instance, to portion pixels, megahertz, and registers in memory. As a mathematical entity, numbers are the objects of the many types of counting that computers carry out: counting of amounts, of sequential steps, of variables, of inputs, of time, and so on. As an incommensurable quantity, numbers approximate the risk of infinite loops within recursive functions. As a digital representation, they mirror electronic binary states into binary digits. As a symbol, numbers are elements of codes and scripts, and cement the possibility of encryption, while also being means of organizing, prioritizing, and enacting such qualities and behaviors.

6. Limits

Computation is limited, in a quite fundamental way. Its limitations are inherent in the axiomatic nature of computational systems. In mathematical logic there are undecidable propositions; in computability theory there exist problems that cannot be solved via purely mechanical procedures. The formal notion of computation itself is founded upon the discovery that some programs will not halt and some functions will not be calculated. Some things just cannot be computed.

The existence of limits in computation is unsettling but also empowering. Amongst the most troubling consequences of these limitations is the comprehension that while computing machines do indeed process many tasks, they do not process just anything. Techno-cultural agendas proposing an all-embracing and all-solving computational rationality are thus faulty at their very outset. Errors, bugs, and glitches might be more or less probable, depending on the specific case. Yet they are always logically possible, as is a formalist misapprehension of a situation or a condition.[13] One of the most interestingly enabling outcomes of the limits of computation, however, results from turning this internal failure into the very method through which computation operates. Limitations, just as with the previously discussed principle of universality, are established at the conceptual level of the computing machine: they are intrinsic to the axiomatic character of computational formalization. Given the necessary provisions, the formal deductions of computational systems nevertheless have been turned into systems of unprecedented instrumental power. To the cultural theorist, the philosopher, and the artist, such mismatches and ambiguities surrounding promises of delivery and potentials for machine breakdown or misrecognition offer an equally finely textured occasion for speculation and are also one of the qualities of computation gamed in the exercise of power.

7. Speeds

Art stages different relations to time: for instance, in the way a dance slows, speeds, accentuates, draws attention to the miniscule or raises it to the level of the cosmic. A relation to, modulation, and creation of time and timings characterizes a work and articulates its mode of being in the world. Computational aesthetics enters into relation with such articulation by intervening in time in certain ways.

The intensity of computational speed is characteristically emphasized as being core to its novelty and to its world-making capacities. When the audience is supposed to pay attention to a rapidly unfolding complex process in the film *The Matrix* (1999), for instance, the scene is rendered to film in great slowness, as if to suggest that—in order to yield something comprehensible to the human sensorium—what passes in less than a moment in computational terms must necessarily be drawn out over minutes. Computational speed is thus about experiential intensity as much as it is about strict measure; yet it is also about the mobile threshold of the capacities of computing itself of structuring its own modes of existence. The speed of calculation of a computer was, from the very outset, in monstrous disproportion to the capacities of humans, just as machine weaving had been to the movements of earlier generations of hand-weavers. This scale of disproportion is fluid, and forms a complex set of texturing of expression that manifests in anything from interaction times in musical instrumentation to media-archaeological concerns regarding speed of execution or bandwidth in the conservation of aesthetic objects.

This issue connects to a subsequent characteristic of speed within computational aesthetics: namely, its constructivist nature. As a consequence of the Universal Turing Machine, the timing of many kinds of processes can be brought about within the machine. Computing has no inherent "native" speed but provides means of staking out and arranging relations to speeds. While intensification of speed is one mode in which computational expression is staged, extension of a work over time is also a significant tendency, especially in works such as Jem Finer's *Longplayer* (2000–)[14] and Gustav Metzger's proposed *Five Screens with Computer* (1965)[15] where the unfolding of a work is arrayed in relation to monumental time. Finer's project, currently sited in London's Trinity Buoy Wharf, assembles a system for making a non-repeating piece of music play for a period of exactly one thousand years. Metzger's plan was to erect a run of five 30 x 40 foot panels of stainless steel 25 feet apart. Each panel would be two feet deep and constructed from 10,000 picture elements of plastic, glass, or steel that would each be ejected over a period of ten years, eventually resulting in the annulment of the work.

8. Scale

Scale effects are core to the development of computing in the present era. By scale effects we mean the ways in which a specific kind of structure or process can be instantiated both across a relatively small number of instances and for a tendentially infinite quantity of them. Systems designed to navigate and conjoin multiple scales, such as the golden mean or Corbusier's Modulor (1954),[16] exist within numerous aesthetic forms as a means of arranging elements or of making and assisting a judgment about these forms' efficacy or beauty. Computing, however, allows for these systems of judgment and composition to be integrated into the technology in which these systems themselves are realized.

In systems such as the World Wide Web, limitations of scale due to material considerations tend toward the negligible, resulting in the use of the term "scale-free" to describe the Web's patterns of network growth. The specific qualities of the scale-free nature of such a system contrast with other aspects of computational aesthetics. This is due to the quality of universality systems such as finite state machines, which have a very small scale that can coexist alongside systems of larger scales. Most notably, however, there may be transitions across the scales. A multitude of small-scale finite state machines, for instance, can be conjoined in order to generate a system of great complexity that can be described as scale-free. Computational aesthetics then includes, as a core constituent, the movement in and out of scalar reference. Concomitantly, the tendency toward a scale-free aesthetics in certain works operating on the basis of these networks can be observed and is to be expected in the future.

The scale-free nature of certain computing forms is coupled in dynamic ways with currents such as globalization, which was explored in early Internet art by Shu Lea Cheang in *Net Nomad* projects such as *Buy One Get One* (1997).[17] Artists such as Ai Wei Wei have used this condition in creating a global constituency for their work, in a period in which the artist—perhaps due to the demise of the reputation of figures such as the politician and banker— also becomes a potential candidate for the role of "moral hero." Other artists, such as The Yes Men,[18] Übermorgen,[19] or Paolo Cirio[20]—whose work is discussed in more detail in Armin Medosch's text—have used the unstable conditions implied by these scale-free networks and globalization as a departure point for exploring sociotechnical expressions of networked forms as they mesh with various political and institutional configurations. Paolo Cirio's loophole4all.com (2013), for instance, is a web site and service that allows users to select the names of tax avoidance entities nominally framed as companies or trusts legally located in the Cayman Isles. Once selected, since these entities need to keep their status shady, the project suggests that their names and tax-free status can be used for invoicing by citizens, thus ensuring the same fiscal opportunities to a wider range of users.[21]

The question of scale is also linked to the development of platforms for cultural expression, since the potentially scale-free nature of a project in technical respects aligns with the capacities of expression of specific social and cultural forces and the individual histories embedded in them. The development of platforms such as the video analysis and combination site Pad.Ma (2008–)[22] by a coalition of groups based in Mumbai, the early picture-sharing platform Nine(9) (2003),[23] or the text discussion site aaaaarg.org [24]—in their combinations of invention and specificity and their amalgamation with other groups, histories, and resources—exemplify such a condition.

9. Logical Equivalence

In earlier discussions of digital media much attention was paid to the question whether a particular experience or thing qualified as "real" or "virtual." Recognizing the medium specificity and the constructivism inherent to computational aesthetics suggests that it might be more fruitful to pay attention to the discussion of forms of logical equivalence.

A system can be said to be logically equivalent to another if it yields the same behaviors and functions—independent of each system's underlying material structure. Logical equivalence is a quality that is foundational to computing as a thought experiment, arising out of the need to equate computing activity with the mental processes of a person. Alan Turing (1936) describes the procedure of making a calculation as a series of mental processes and note-making, abstracting the procedure to a formally describable set of steps that can then be instantiated in a machine. The result is an axiomatic procedure that can be universally applied to any computable problem (the a priori limits of the axiomatic procedure itself, however, remain intractable). Simulation is one effect of establishing logical equivalence between systems. An entity or process may be ordered in such a way that it is rendered more or less behaviorally identical to that which it models. At the same time, there may be play within the kinds of equivalence that are operative. At a certain scale, for example, a system may display logical equivalence to another, yet be composed of substantially different materials. There also may be interplay with the subjective experience of each different instantiation of a logically equivalent event or performance. The musicological concept of interpretation may be pertinent here. The translation of behaviors and entities from other contexts into computational ones implies an evaluation of what constitutes meaningful forms of equivalence and thereby the intensification of aesthetic, along with ethical, judgments. The interplay of these conditions has proven to be very fertile ground for exploration for many artists.

In his ongoing *Status Project* (2005–), for instance, Heath Bunting sets out to establish a logically equivalent description for the process of attaining membership in various kinds of social formations (such as nation states or video libraries). Being a certain age, having an address, a name, being able to produce and refer to other specific documents—by following certain set and delimited procedures, one may acquire a position that can be computed as verifiable within a logically describable system of veridiction, a statement that is true according to the worldview of a particular system. The condition of logical equivalence has also driven much work in the field of bio art, where the reduction of the characteristics of DNA to its base pairs TCAG (T = thymine; C = cytosine; A = adenine; G = guanine) allows for the rearticulation and the handling of the amino acids that they signify and partially render tractable. In the project *The Xenotext* (2011), the poet Christian Bök exploited this context to encode a short poem in a bacteria that would in turn produce protein that would be readable, via the use of Bök's interpretative system, the Chemical Alphabet, as further fragments of poetry.[25]

10. Memory

Within computing, the fact that both data and the instructions that act upon that data are themselves stored as data has significant consequences. Not the least of these consequences is that—since a computational machine can be completely copied with great ease under the conditions of logical equivalence—the conditions for an effective digital commons can be produced. The fact that both a computer and the data that runs on it (including the operative and executable data software) can be copied, creates interesting situations in politics and economics, situations that also have consequences for art and contribute to the social and economic force of computing at large.

Memory also introduces other conditions and forms of computational aesthetics: possibilities both for all actions and interactions to be logged in order to be restaged or analyzed, and, within different configurations, for the (partial or full) reversibility or irreversibility of an action. Moreover, memory presents conditions of delay and storage, so that an event may unfold in computational time at a different moment. Related to the question of speed, time—as it manifests in the interrelation between processing and storage and in the interaction between the computational system and a subject, such as a musician, dancer, or game-player—becomes a crucial factor in the developments of the aesthetic modality of both specific systems and computational systems as a whole.

Memory, understood as the extent of the capacity to process data, also has significant effects for digital art. These effects are readily observable in 8-bit aesthetics (Goriunova 2012), where constrained data-architectures are adopted or simulated for the pleasures of their nostalgic and simplified forms. However, they can also be seen in the use of any constrained computing system—that is to say, any at all. We can also argue that memory is exemplified in an interplay between learning and the relentless lack of it in computing. Kryštof Kintera's human-scaled robot *Revolution* (2005) (Horáková 2012) beats its hooded head endlessly, over and over, against a wall. This is not a generative error, as a glitch would be, but a machinic ability to repeat without recourse to reflection.

If – then

At this point any reader will probably have added some other aspects of computational aesthetics to this list. Our aim is not to be complete here. Indeed, part of the question of the aesthetics of contemporary digital media, particularly as they are developed by artists, is to advance and proliferate frameworks for recognizing further modes of existence for the computational. The task of doing so is a collective one, and cannot be reduced to a schematic list of qualities, or to a set of conditions imported into art directly from an understanding of different forms of computer science. Once again, we would like to stress that computational aesthetics, and the immanent investigation of it, reside in the computational event. Computing and its aesthetics are no longer "owned" by the disciplines and fields that grew up closely in and around it. The computational mundanity of everyday objects and processes, as well as the more explicitly critical and speculative modes of computational forms, may be interrogated by means of the characteristics that we have discussed above. At the same time, the nature of the computational may be changed altogether by bringing more conditions and forms of existence into its purview. All of this together, along with the very flexibility of computational reality, means that these considerations can only ever be a provisional and partial mapping. There is much to invent and to be dazzled by, much texture to be found. One might also discover a strange, dull, as yet unnameable familiarity to certain repetitions and compulsions that may indeed travel unremarked from art installations to office work and social forms. To go beyond such a list means to engage in a preliminary process of recognizing and operating the aesthetic dimensions of computation. As critical experimental work moves more substantially in this direction, the force and method of computation may become more open to understanding, discovery, and invention.

Notes

1 For instance, Stiegler (2012) argues that the irreducible ambivalence of technological rationality is altering all forms of knowledge, and thus "we must learn to think and to live differently."

2 Although such conditions of internality and externality may also be part, or indeed be imperative to, aspects of the computational method (as with the specific forms of modular architectures, object-oriented environments, the limited modes of abstraction layers such as interfaces, and so on).

3 The history of art is, in some respects, that of an interplay between the abstract and the concrete, as they are understood and made manifest by different means over time. In a sense, we live at a moment in which the abstract itself, as a force and a method, is understood to be of highly diverse character. The modes of abstraction in art, having generated a history of significant range, now also manifest this proliferation, the consciousness of which in turn has its own effects. A significant exploration of this context can be found in the work of Matias Faldbakken (2007), who produces an intensive bestiary of modes of abstraction in relation to the powers of materials.

4 See http://www.mongrel.org.uk/lungszkm/
5 See http://www.mongrel.org.uk/londonpl/
6 See http://yoha.co.uk/cfc/
7 See http://www.post-data.org/beige/
8 See http://post-data.org/beige/abstract.html
9 See http://www.davidrokeby.com/
10 See http://www.davidrokeby.com/nchant.html
11 See http://piratbyran.org/
12 Here we are referring to the "ethico-aesthetic paradigm" of Guattari (1995). Guattari draws the expression from Mikhail Bakhtin, and uses it to denote the way in which collective subjectivity can, through the techniques and practices epitomized in (but not limited to) art, constitute and project itself toward alterity and heterogeneity. For Guattari, aesthetics has ethical and political implications, in so far as "to speak of creation is to speak of the responsibility of the creative instance with regard to the thing created, inflection of the state of things, bifurcation beyond pre-established schemas, once again taking into account the fate of alterity in its extreme modalities" (1995, 107).
13 This narrow or voluminous gap is, for instance, that occupied by the discussions of so-called ethicists in their prevarications on the operation of automated warfare, such as that carried out by drones.
14 See http://longplayer.org/
15 See Ford (2003).
16 Le Corbusier's Modulor is a scale of proportions that was based on the golden ratio and developed on the model of the human body.
17 See http://www.ntticc.or.jp/HoME/
18 See http://theyesmen.org/
19 See http://www.ubermorgen.com
20 See http://www.paolocirio.net/
21 See http://loophole4all.com/
22 See http://pad.ma/
23 See http://www.mongrel.org.uk/nine
24 See http://aaaaarg.fail/
25 See http://www.poetryfoundation.org/harriet/2011/04/the-xenotext-works/

References

Aaaaarg.org. http://aaaaarg.fail/ (accessed October 15, 2015).

Beige. http://www.post-data.org/beige/ (accessed January 4, 2015).

Beige. 2000. *Super Abstract Brothers*. http://post-data.org/beige/abstract.html (accessed January 4, 2015).

Bök, Christian. 2011. *The Xenotext*. http://www.poetryfoundation.org/harriet/2011/04/the-xenotext-works/ (accessed January 4, 2015).

Bossa Studios. 2013. *Surgeon Simulator*. http://www.surgeonsimulator2013.com/ (accessed January 4, 2015).

Bunting, Heath. 2005– . *The Status Project*. http://status.irational.org/ (accessed January 4, 2015).

Chaitin, Gregory. 2005. *Meta Maths: The Quest for Omega*. London: Atlantic Books.

Cheang, Shu Lea. 1997. *Buy One Get One*. http://www.ntticc.or.jp/HoME/ (accessed January 4, 2015).
Cirio, Paolo. http://www.paolocirio.net (accessed January 4, 2015).
Cirio, Paolo. 2013. *Loophole For All*. http://loophole4all.com/ (accessed January 4, 2015).
Davis, Martin. 2000. *Engines of Logic: Mathematicians and the Origin of the Computer*. New York: W.W. Norton.
Deleuze, Giles, and Felix Guattari. 2004. *A Thousand Plateaus: Capitalism and Schizophrenia*, translated by Brian Massumi. London: Continuum.
Faldbakken, Matias. 2007. *Not Made Visible*. Zurich: Christoph Keller Editions.
Finer, Jem, 2000– . *Longplayer*. http://longplayer.org/ (accessed January 4, 2015).
Ford, Simon. 2003. "Technological Kindergarten." *Mute* 1(26), July 4. http://www.metamute.org/editorial/articles/technological-kindergarten (accessed January 4, 2015).
Fuller, Matthew, ed. 2008. *Software Studies: A Lexicon*. Cambridge, MA: The MIT Press.
Goriunova, Olga. 2012. *Art Platforms and Cultural Production on the Internet*. London: Routledge.
Greenberg, Clement. 1961. *Art and Culture*. New York: Beacon Press.
Guattari, Felix. 1995. *Chaosmosis: an Ethico-Aesthetic Paradigm*, translated by Pail Bains and Julian Pefanis. Bloomington, IN: Indiana University Press.
Harwood, Graham. 2004. *London.pl*. http://www.mongrel.org.uk/londonpl/ (accessed January 4, 2015).
Harwood, Graham, with Mongrel. 2005. *Lungs (Slave Labour)*. http://www.mongrel.org.uk/lungszkm/ (accessed January 4, 2015).
Horáková, Jana. 2012. "The Robot as Mitate: The Media Specific Aesthetic Analysis." In *Device_Art,* festival catalogue, 60–89. Zagreb: Kontejner.
Le Corbusier. 2004. *The Modulor: A Harmonious Measure to the Human Scale Universally Applicable to Architecture and Mechanics*. Basle and Boston: Birkhäuser.
Manovich, Lev. 2013. *Software Takes Command*. London: Bloomsbury.
Maturana, Humberto, and Francisco Varela. 1992. *The Tree of Knowledge: The Biological Roots of Human Understanding*. Boston: Shambhala.
Mongrel. 2003. *Nine(9)*. http://www.mongrel.org.uk/nine (accessed January 4, 2015).
Pad.ma. 2008– . http://www.pad.ma/ (accessed January 4, 2015).
Pattinson, Yuri. 2013. "e ink pearl memory." In *Open Office Anthology* edited by Tom Clark, Farkas Rozsa, and Harry Burke, 58–61. London: Arcadia Missa.
Piratebyrån. http://piratbyran.org/ (accessed October 15, 2015).
Rokeby, David. http://www.davidrokeby.com/ (accessed October 15, 2015).
Rokeby, David. 2001. *n-Cha(n)t*. http://www.davidrokeby.com/nchant (accessed January 4, 2015).
Stiegler, Bernard. 2012. "Die Aufklärung in the Age of Philosophical Engineering." *Computational Culture*, 2. http://computationalculture.net/comment/die-aufklarung-in-the-age-of-philosophical-engineering/ (accessed October 2, 2015).
Turing, Alan M. 1936. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society* 42: 230–265.
Übermorgen. http://www.ubermorgen.com (accessed January 4, 2015).
Yes Men. http://www.yesmen.org (accessed January 4, 2015).
YoHa, with Jean Demars. 2006. *Coal Fired Computers*. http://yoha.co.uk/cfc/ (accessed January 4, 2015).

A Companion to Digital Art

Edited by

Christiane Paul

WILEY Blackwell

This edition first published 2016

Registered Office
John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Offices
350 Main Street, Malden, MA 02148-5020, USA
9600 Garsington Road, Oxford, OX4 2DQ, UK
The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

For details of our global editorial offices, for customer services, and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com/wiley-blackwell.

Library of Congress Cataloging-in-Publication Data

Names: Paul, Christiane, editor.
Title: A companion to digital art / Edited by Christiane Paul.
Description: Hoboken : John Wiley & Sons Inc., 2016. | Includes bibliographical references and index.
Identifiers: LCCN 2015039613 (print) | LCCN 2015040809 (ebook) | ISBN 9781118475201 (cloth) | ISBN 9781118475188 (Adobe PDF) | ISBN 9781118475218 (ePub)
Subjects: LCSH: Computer art.
Classification: LCC N7433.8 .C615 2016 (print) | LCC N7433.8 (ebook) | DDC 776–dc23
LC record available at http://lccn.loc.gov/2015039613

A catalogue record for this book is available from the British Library.

Cover image: Scott Snibbe, *Deep Walls*, interactive video installation, San Francisco Museum of Modern Art, 2002. Reproduced by permission of Scott Snibbe.

Set in 10/12pt Galliard by SPi Global, Pondicherry, India

1 2016