SOFT COMPUTING: FORMS AND LIMITS IN COMPUTATIONAL AESTHETICS

M. Beatrice Fazi

This paper contends that soft computing can help us investigate the aesthetics of digital computation. Employing broader conceptions of aesthetics and perception, and whilst drawing upon the ontology of Alfred N. Whitehead, it uses soft computing to address the 'prehensive' dimension of the quantitative procedures of computation, and explores the interrelationship between the factuality and formality of computational structures.

This paper will argue that soft computing may be of relevance to the field of computational aesthetics. By focusing on the theoretical foundation of aesthetics in computation, I will show that soft computing can highlight the indispensible role played by abstract processes within the construction of experience.

Soft computing is a rapidly advancing area within computer science. It is characterised by an attempt to deal with uncertainty, approximation, randomness, and partial truth, and some of its most prominent examples include neural networks, fuzzy logic and evolutionary computation. 'Soft' techniques differ from traditional computing as they employ diverse methodologies in order to cope with the difficulties involved in achieving and maintaining algorithmic efficiency. Their originality and appeal lie in their ability to provide inexact, indeterminate and generative solutions to 'computationally hard' problems (i.e. questions that are too complex to be addressed via classic computational systems).

I will contend that soft computing can offer a means of investigating the aesthetic dimension of digital computation. However, 'aesthetics' will be understood here in a manner that exceeds the disciplinary bounds of a theory of art or beauty: in keeping with its etymological roots ('aisthesis'), it will be taken to denote a theory of relationality and perception. One further qualification: 'perception' will not be viewed as referring solely to human cognitive faculties. Just as aesthetics will be understood in a non-anthropocentric sense, so too will perception: it designates the ways in which experience is constructed, thus referring to the manner in which we encounter things and to the ways in which these things encounter others in turn. Consequently, 'prehension' may be a more appropriate term than 'perception.' I take this suggestion from the work of the philosopher and mathematician Alfred N. Whitehead, within whose ontology prehension plays a special role: it refers to the most foundational type of relation – the extrasensory awareness that all actualities have of earlier and future occasions – and is used to explain how all entities within the realm of actuality are acts of experience. Such experience is not the exclusive domain of human consciousness. All actualities are in fact held to experience one another, whether they are conscious or not, as the data that they inherit from past occasions leave them internally connected. [1]

Whitehead's ontology and the broader notions of aesthetics and perception that I have drawn from it allow us to view computational structures as possessing a relational dimension that exceeds both their phenomenical effects and the intentionality of their users. My contention is that computational structures can be understood as systems of actual occurrences: they are concrete 'facts,' fully realised actualities, particular to their own spatial and temporal occasion. To focus only on the performative and qualitative character of these events, however, is insufficient. In my view, factuality itself is enabled by levels of quantitative, logico-mathematical abstraction. In a computer science context such abstractions are usually understood as computing's modus operandi: those methods or processes that follow a well-defined procedure and describe how a task is to be performed. Yet I would argue that logico-mathematical abstraction also constitutes the 'form' of computation itself, i.e. its potential character, the pattern that defines its ontological possibility. My interest here is in presenting computational structures both as facts (actual instances, events) and as forms (patterns of potentiality). The aesthetic relevance of computation, in my view, lies in the irreducible relationship between the two. I will show that soft computing practices emphasise this interrelationship of fact and form, and thereby highlight crucial, although largely unaddressed, issues within computational aesthetics.

The relevance of this approach can be illustrated by looking at current understandings of computational aesthetics, but in order to do so I need to make a brief qualifying remark on digitality. Computation and digitality are by no means synonymous, yet both can be understood as processes of discretisation: computation, by virtue of its axiomatic character, involves the discretisation of procedures; likewise, digitalisation can be seen as the technological automation of those discrete procedures. I will focus on the discrete aspects of both computation and digitality below, as in doing so we can underline the fact that contemporary computational aesthetics has overlooked the prehensive dimensions of quantities.

This is important, as the concept of prehension affords a relation between the discrete and the continuous that has eluded contemporary computational aesthetics' attempt to connect the digital to the analogue. Media and cultural theorists, computer scientists and philosophers still disagree on an exact definition of digitality, but they do seem to view the digital as a discrete data technology that uses discontinuous values to access, represent and manage information. Yet, at the same time, the vast majority of philosophical, cultural and social accounts of aesthetic experience portray a universe of percepts and perceivers, the reciprocity of which is established by a rapport of continuity with what is given in experience. Hence the difficulty that digital aesthetics seems to be faced with: where the digital tends to be understood in terms of the discrete, aesthetics tends to be understood in terms of the continuous, or the analogue. The discrete nature of numerical and data quantities is thus largely omitted from aesthetic considerations of the digital medium. These accounts are often geared towards analysing its perceptual, performative, and phenomenical effects, which are possessed of characteristics often associated with the analogical. Codes, scripts, values, parameters and algorithms are viewed as performing actions, which in turn exhibit qualities and properties. Agency and quality thus appear to be key to the disclosure and employment of the aesthetic value of computation. These approaches risk dismissing the mathematical and logical nature of computational digital media. If the latter is specifically addressed, it is done so in a manner that understands the ontological power of logico-mathematical abstraction only in terms of continual calculation; this tends to be characterized as 'topological' in nature and expression, and flattened onto a plane of differential transformability. I, however, would like to propose that the impasse between continuity and discreteness could be re-thought in the light of the prehensive dimension of computational structures, as prehension exists before and beyond any performance, action, or effect. The approach that I am arguing for would retain the possibility of working with the reality of algorithmic entities by addressing the aesthetic dimension of their quantitative, discrete procedures.

Soft computing can help us explore these issues, for it serves to highlight that the rules and quantities of algorithmic construction are always involved with patterns of potentiality, and that these patterns are themselves expressed by abstract logical processes. It thus illustrates the importance of the interrelation of factuality and formality within computational structures. There are a number of ways to prove this speculative hypothesis, but I will attempt to do so here by looking at the empirical and rational characteristics of soft computing methodologies.

As noted at the outset, soft computing uses uncertainty and randomness to solve problems that humans are adept at, but which classical computational methodologies struggles with (face recognition or linguistic disambiguation, for example). Whilst indeterminacy is considered undesirable in the classic theory of computation, soft computing exploits it in order to obtain tractability, lower solution cost and a certain economy of communication. The metaphoric 'softness' of this approach is meant to afford an alternative to the 'rigidity' of the conventional analytic methods that found the classical theory of computation. Soft computing can thus appear more tuned to the 'empirical' levels of reality than traditional computation, as it seems to allow factual chance into its algorithmic formality. This point is perhaps proved by its ability to solve tasks that involve information gained from or modified by experience, such as the industrial application of intelligent systems geared towards solving 'real life' tasks (e.g. control, modelling and simulation). Soft computational systems have also been utilized in social science contexts, where adaptation to imprecise judgment, sense perception and emotion is key.

Generative algorithms provide an example of soft computation's relation and frequent association with the empirical. Chance, I would contend, is particularly important for generative algorithms, as they encode formal rules via the application of a bottom-up approach designed to encompass contingent change over time. The programmer deliberately writes very simple instructions, and lets complex behaviour emerge through an iterative selection process, which picks the best representations of solutions, rejects bad results, and produces new ones from those that survive the procedure. Generative algorithms are therefore able to modify their own code, allowing new combinations to arise from parallel and random behaviour, and thereby simulating natural selection and biological evolution. [2] This process fosters autonomy and uniqueness, and has in consequence been employed by artists as a means of playing with the underlying rules of formal generation and structure. It however also offers an interesting prompt for us to speculate as to the ways in which empirical factors – in the guise of variation, selection and evolution – can enter the formalism of computation.

The example of generative algorithms illustrates that soft computing can be seen to be characterised by an empirical aspect. Yet this does not mean that soft computing dismisses logic and formality: where others have emphasised its orientation towards the empirical, I would argue that its most interesting theoretical implications lie in its indirect continuation of what might be called a 'rationalist' project of optimization, compression and synthesis.

My point is that soft computing, despite its openness to contingency, still operates within the rational, disciplinary bounds of mathematics and logic. This can be seen in fuzzy logic systems. Whilst the traditional method of computation uses binary logic, which permits only a dichotomous opposition of true and false, fuzzy systems compute via a logic that allows for differing truth values: they are thus able to reflect the imprecise definitions of language, and can engage with complex control, management and recognition problems that cannot be framed in 'crisp' terms. The fundamental idea behind fuzzy logic is that all things exist in degrees, and that we are in consequence bound to encounter imprecision in truth. Fuzzy logic responds to this (as does the seminal fuzzy set theory first outlined in 1965 by Lofti Zadeh) [3] by trying to tune knowledge representation, so as to make artificial systems function more like humans. Like the generative algorithms described above, fuzzy logic thus accommodates the contingent conditions of experience, but in capturing loosely-defined categories and generalising them it remains a formal model; so too does the generative algorithm, which puts forth a problem-solving strategy in terms of a set of axiomatic rules evolved upon a specific yet formally addressed situation. In sum: although many have associated the merits of soft computing with this orientation towards the empirical, I would argue that it is this formal dimension that constitutes its real potential.

This claim can be developed by looking further at fuzzy logic. The latter is a logical calculus designed to build a formal system capable of handling information devoid of analytic formulation. Fuzzy logic, however, also produces a logico-mathematical structure that can be computed, and which can be put into the finite terms that machines require in order to operate. One can therefore comment that the rationalizing power of soft computation stems from its capacity to enlarge the realm of actuality, through accommodating factors that could not be encompassed by mathematical formalization. Soft computing has consequently been viewed as incorporating a degree of quality into the computational process, as it can be interpreted as a quest to capture the 'continuity' of things through approximation. Imprecision would then seem to arise from a continuum possessed of degrees that can be broken down into 'fuzzy' chunks. One can see how this project can perhaps be associated with the 'analogical' digital aesthetic perspectives discussed above, which frames aesthetics in terms of the continuous transformation of perceptual qualities. Yet, in my view, fuzzy logic is also characterised by attentiveness to the implicit rationality of the real: its focus rests not on binary calculation per se, but rather on the inherent possibilities of calculation itself. Thus, contra those who would claim that the importance of soft computing lies in its attention to quality, I would argue that it returns us to the significance of the quantitative in computational aesthetics.

Fuzzy logic's advocates often emphasise that the technique uses imprecise indicators instead of exact values. However, I believe that if we look at the manner in which fuzzy systems deal with vagueness we are brought back to a question implied by Alan Turing's classical theory of computation, which he set out in 1936: how could one devise an 'effective method' composed of finite terms that might be able to put the infinity of quantities into the finite terms of a procedure? [4] Such a method would be a process of discretisation, as is the method by which fuzzy logic deals with vagueness. Briefly, fuzzy logic works through a system of sets with graded members; each grade is possessed of a different truth value, and each member thus forms part of a smooth but differentiated continuum. One might read this as an attempt to turn quantity (discrete grades) into quality (varied continuity), and to create a means of computation that comes a little closer to 'reality'. However, I would hold that it in fact remains just as reliant upon the quantitative and the discrete as Turing's theory of computation, as soft computing's ability to cope with the uncertain is itself grounded on the quantitative nature of computational formality. Fuzzy logic doesn't render quantity qualitative by attributing continuous characteristics to it: rather, the mathematical processes that allow fuzzy systems to accommodate approximation are themselves based on the procedural and axiomatic modes of quantitative abstraction.

It is important to stress now that the observations above locate soft computing within a broader debate about the limits of formal reasoning. These limits were envisaged by the logician Kurt Gödel in 1931, [5] and subsequently framed in terms of computability by Alan Turing in the 1936 paper mentioned above. Turing's famous thought experiment, the so-called Turing Machine, established the theory of computation by demonstrating that there are limits to what can be computed. His account has however received criticism over the last two or three decades, and much of this has involved framing his model as a closed formalism that doesn't allow for the influence exerted by the external factors of contingent reality upon its purely internal world of algorithmic procedures. This brings us back to soft computing, which as we have seen is engaged in an attempt to 'open' formalism to contingency. Yet, in my view, it is not by introducing the empirical that we are afforded a more 'open' formalism; rather, this openness is already granted by virtue of the fact that computation has intrinsic limits. I would contend that Turing cast uncertainty and randomness as intrinsic features of his model of computation by discovering the notion of the uncomputable, as this means that computation becomes defined by that which it is not: the formal logic of the algorithmic method is consequently always already 'open', as it tends towards its own limits.

We are thus returned to the import of the quantitative in computational aesthetics. We have seen that soft computing attempts to introduce the empirical into its calculations. Yet we have also seen that it does so without fully abandoning the rationality of the formal method. So, rather than focusing on soft computing's accommodation of contingency, I would argue that we should instead take soft computing as an illustration of the manner in which quantity is always already involved in the qualitative. This point can be made on two levels: firstly in terms of computer science, and secondly by way of reference to the philosophical ideas sketched at the outset of the paper. We should note that computing systems function through discrete processes: through axiomatic structures that operate via finite, and thus quantitatively distinct steps. Quantity also grounds the logical forms of computational structures, as it defines and shapes their possible configurations. Yet in Whitehead's ontology, as described above, reality itself is a succession of actual facts. Each is informed by a prehensive relation with other facts, and each has an equally prehensive relation to the quantities of their potential patterns. Earlier I described such patterns as the forms that enter into these actual facts. If we now consider these patterns of potentiality as the logical and mathematical forms of computational structures, then we would have a means of according computational aesthetics a far broader reach than has commonly been ascribed to it.

In conclusion, I believe that one of the most interesting issues that can arise from this way of thinking about soft computing is an awareness of the mutuality between factuality and formality. Soft computational structures are factual, as they are computational events with an actuality. In other words, they are acts of experience, as they inherit their constitution through the prehension of other acts of experience. This is their factual existence which, by virtue of this relationality, is possessed of an aesthetic dimension that is not uniquely reliant upon the contention – common amongst the approaches to aesthetics described above – that computational processes exhibit effects, agency and quality. I would also stress here that soft computing structures have a formality, and that this is expressed through their logico-mathematical character. For example, the procedure by which generative algorithms operate varies with contingent change: the factuality of this computational structure thus has a relation with the empirical world. In its factuality, however, it needs also to refer to what we referred to as 'forms' above, i.e. its logico-mathematic pattern, and this relation to ideality can once again be termed prehension. It is here, beyond the phenomenological outputs of computational structures, that we find the real ontological motive force of their interrelation. The forms of computation can thus be seen as an ontological, existent reality, and not just as a type of deductive reasoning that contrasts with the direct experience advocated by inductive approaches. Aesthetics, in this sense, is not the subjective judgement of the perceptual experiences that a computational structure presents to us; instead, it is the relation between factuality and formality. Although soft computing is limited in the degree to which it can help us develop this position, I believe it can help us shift contemporary discourse towards the issues that are indicated here.

References and Notes:

1. *Alfred N. Whitehead, Process and Reality. An Essay in Cosmology (New York: Free Press, 1978).*
2. *See Melanie Mitchell, An Introduction to Genetic Algorithms (Cambridge, MA: The MIT Press, 1998).*
3. *Lotfi A. Zadeh, "Fuzzy Sets," Information and Control 8, no. 3 (1965): 338–353.*
4. *Alan M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," Proceedings of the London Mathematical Society 2, no. 42 (1936): 230–265.*
5. *Kurt Gödel, On Formally Undecidable Propositions of Principia Mathematica and Related Systems, trans. B. Meltzer (New York: Dover Publications, 1992).*