Wolfgang Banzhaf

16 Morphological Computation – A Broad Perspective

Abstract: *In this contribution a broad perspective on morphological computation is introduced. We shall argue that shape is a special feature of spatial relations and shall advocate a computation model that takes into account spatial relationships between components of the computation.*

Current Definitions

When trying to get a clear picture of a certain term, it is often useful to look at the opposite to see whether that term carries any meaning. What would be the opposite to morphological computation? One guess would be ”morphological non-computation," and this term might indeed have some legitimacy, but not in our context. Rather "non-morphological computation" would be the proper choice of words for the purpose of this discussion. Some would say, "amorphous computation," a term that has indeed been used in the recent past [1].

The authors write: "An amorphous computing medium is a system of irregularly placed, asynchronous, locally interacting computing elements. We can model this medium as a collection of "computational particles" sprinkled irregularly on a surface or mixed throughout a volume. [...] Each particle has modest computing power and a modest amount of memory. The particles are not synchronized, although we assume they compute at similar speeds, since they are all fabricated by the same process. The particles are all programmed identically, although each one has means for storing local state and for generating random numbers. In general, the particles do not have any a priori knowledge of their positions or orientations."[1]

In other words, amorphous computing is massively parallel, assumes identical elements (at least ideally) of limited capability and homogeneity of substrate, no a priori knowledge of location of elements relative to each other, but emergence of behaviour from local interactions. Local interactions are perhaps best described as a requirement of the massive parallelity in these systems. The identity of elements is a vestige of engineering thinking, which in the real world can only be an approximation. Overall, the aim of amorphous computation is algorithmic abstraction.

Let us now turn to a definition of morphological computing / computation. In [14] Rolf Pfeifer et al. emphasize that there are information-theoretic consequences of embodiment of agents in the world. They write: "By information theoretic implications of embodiment we mean the effect of morphology, materials and environment on neural processing, or better, the interplay of all these aspects. It turns out that materials, for example, can take over some of the processes normally attributed to control, a phenomenon that is called "morphological computation." There is no taxonomy of morphological computation yet, but we can roughly distinguish between sensor morphology taking over a certain amount of computation, similarly for shape and materials, and for the interaction with the environment." [14, p.23]

These authors thus emphasize the aspect of physical embeddedness which all agents "suffer" that are embodied. This refers to the physical relation between the agent's components, their individual material properties, and their physical interactions with the environment. The authors argue strongly (and rightly) that physics holds advantages in terms of providing certain computation for free, computation that would otherwise require substantial effort. They cite space-variant vision [8] as one example where the inhomogeneous placement of light-sensitive cells supports computational vision tasks in sensory information processing. In another example, the authors discuss motor information processing by applying a physical spring as an artificial muscle. Summarizing, they

Feature	Morphological Computation	Amorphous Computing
Elements	individuals	clones
Organization	heterogeneous	homogeneous
Interaction	local and global	local
Space	localized	non localized
Approach	applied, concrete	general, abstract

Table 1: *Contrasting Morphological Computation with Amorphous Computing.*

state: "The morphological computation in this case is the result of the complex interplay of agent morphology, material properties (in particular the "muscles," i.e. the springs), control (amplitude, frequency), and environment (friction, shape of the ground, gravity). Exploiting morphological computation makes cheap rapid locomotion possible because physical processes are fast and for free!" [14, p.25]

In other words, morphological computation is very much concerned with the materials that do the computation, their physical properties, but also their form, i.e. spatial relationships of components to each other. Generally speaking, such systems are to be expected to be heterogeneous, and rather than being concerned with the abstraction of computational processes, these systems exist or are designed/evolved for a particular purpose or function. Efficiency criteria play a major role and abstraction has taken a backseat. However, as we shall see, the current notion of morphological computation falls short: It does not include the centre of information processing in a body, rather only input and output devices. Below we shall therefore propose a generalization of the concept of morphological computation that encompasses a central information processing unit.

As we can see: A resounding contrarian view to amorphous computation which can be summarized in the Table 1.

The Relation between Morphology, Space and Nature

In a more recent publication by a group including Pfeifer [11] it is argued that the complexity and non-linearity provided by flexibly moving body parts of a robot - while difficult to control - is a source of computational power and therefore a desired feature.

In a series of recent publications Gordana Dodig-Crnkovic has taken up the idea of morphological computation and built a framework called "info-computationalism" [5, 6, 7]. Within this framework of a new philosophy of computing, morphological computation is at the centre of all methods of natural computation.

It is interesting to note that the relation between morphology, space and Nature was important already earlier in the history of the sciences. In the 19th century, it stood at the cradle of a new discipline: geography. It was Alexander von Humboldt, who, through a strike of genius, was able to connect the morphology of a landscape with its topology, and therefore with space, founding the discipline of geography at the intersection between physical sciences and the humanities [12]. Are we witnessing a similar phenomenon now in regard to computation?

Morphology always presupposes a notion of topology and space, otherwise shape and relation between parts could not be defined. Space, in turn, is intimately connected to Nature. Nature without space is unimaginable, therefore, we associate the beginning of natural systems, and indeed Nature, with the beginning of space as is reflected in the standard model of the universe [2].

Note that most of what has been hitherto considered computation in the literature does not require a notion of space, perhaps with the exception of cellular automata [4, 15]. Some have, however, argued in turn, that the universe is a calculating machine [9, 13], based on notions of cellular automata and thus based on the only model of spatial computation known at the time.

Morphological Computation - A Broad Perspective

In recent years, computer scientists have realized that the world is more complex than what the notions of a Turing machine or a von-Neumann machine (a clever embodiment of the concepts of computation promoted by Turing) suggest. As Dodig-Crnkovic points out, these notions are typically closed system models, isolated boxes in which "computation" happens determined by a static input, a machine in defined states and an algorithm that should be left alone until a result emerges. The emphasis on the halting problem in classical theoretical computer science is a typical outcome of this thinking. Sure enough, if the algorithm doesn't halt, no defined result emerges, and the whole computation has a problem.

Contrast that with what is now called data stream processing [3], a model of computation where input consists of continuous, unbounded, rapidly changing streams of data elements. Such a model of open, dynamically changing, massively parallel data streaming into a computational system is perhaps better suited to capture the interaction of a living organism with its environment than if we try to formulate algorithms for a closed systems. What it requires is a constantly running computational engine, because the data streams also never cease to flow. These open computational systems need to have a means to store relevant information extracted from the data stream in a finite storage. Despite the infinity of the stream, a way must be found to store information extracted from the stream in a finite memory. Most of the stream must leave the system after it has contributed to some computation and will have to be discarded ultimately, due to the impossibility of accumulating the content of the stream. In fact, a better metaphor would be to say that the computational engine must sit at the "shore" of the data stream and dip into it as required.[1]

[1] This would mean the machine would merely observe the data stream, without "swallowing" it. This would entail the necessity of a translation of these observations into an internal representation of the machine, and would allow different machines to produce different representations/extracts/results from the same data stream, see [11] for a similar argument about the usefulness of such a property. In particular, this could open the door for a competitive evolutionary process to improve information extraction from the data stream, just like what happened with the evolution of brains.

Hauser et al. [11] make a similar argument as regards the "morphological computation with compliant bodies." They claim that for biological systems a mapping from input streams to output streams (which could be formalized by mathematical operators) is closer to real demands than a Turing machine. Our expectation is that this is a fundamentally different model of computation which requires attention from computer scientists for its potential to one day eclipse the Turing/von-Neumann paradigm.

Such a fundamental shift will probably come about by the practical needs of applications. In the area of data streaming we are witnessing already the emergence of such needs . A few examples of typical data streaming applications are:

- Network traffic streams
- Financial data streams
- Sensor network data streams
- Weather observations
- Measurements and observations from physical systems like particle accelerator data or telescope data
- Sensor input streams for robot navigation
- Activity streams in online computer games

It is interesting to note that the idea of data streams was discussed first in the database community, before finally spilling over into other areas of computer science. The question originally was how to manage data streams, as they obviously constituted challenges to the storage capacity of database systems. So query systems for a data stream management system (DSMS) have been prevalent in researchers' minds [10].

While it is certainly important to study how to manage data streams, the more important question in our context is whether the principles of computation applied to streams of data would not need to be radically changed as well. I would answer this question in the affirmative. The solution to this problem is morphological computation in the sense that different centres of processing will feed on the data stream, and extract pieces of compressed information in a way prescribed by their type.

Just as the brain consists of many localized centres of information processing of different types, all connected to each other and ultimately to sensors and effectors, so a computing machine could be constructed using localized centres specializing in different types of algorithms, all either connected to each other or, ultimately, to sensors and effectors.

As an example, consider a soft computing machine consisting of localized centers, for instance, a machine learning centre, a genetic search engine, a fuzzy reasoning centre, a swarm intelligence device, and a group of neural networks. We use to think of these entities as different types of algorithms, but in morphological computation they each exist side by side, in a spatially configured computing machine. They are localized,

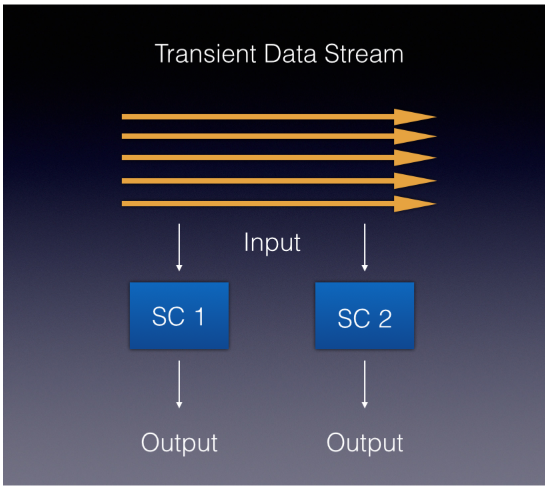

Figure 1: *In this sketch, two soft computing machines (SC 1 and SC 2) dip into a transient data stream and copy information into their internal data representation. Subsequently, internal processing allows them to compute reactions that constitute their output.*

heterogeneous pieces of (possibly different, specialized) hardware and connect to each other. Input to this machine would be from a stream of data, and changes in processing of information would be effected not so much by changes to the different centres, but by different routing of the internal representation of the data stream and its higher-level extracts.

We can see immediately, that this is the idea of morphological computation, this time, however, applied to the computation itself. Modern brains of higher-level organisms are organized precisely in this fashion. The human brain, for instance, has more than 50 nuclei, different subsystems of nerve cells discernible by their internal connectivity structure and the elements performing the computation. Major changes in information processing in brains are achieved not so much by reprogramming particular specialized nuclei, but by changing their connection to other nuclei and the introduction of new nuclei.

In a similar manner, major changes in the soft computing machine sketched above could be achieved by introducing new centres for processing information, as well as by rerouting the information circulating in the system. Thus, programming would be a fundamentally different process. Sensor information streaming into the system would have to be ultimately discarded in favour of the extraction of highly compressed representations of these streams. Figures 1, 2 show a sketch of this idea.

In general, this open model of computation is much more natural and could be argued to be the quintessence of embodied computation. It comes with another advantage: The possibility to allow a multitude of these machines to access the same data stream. Why would this be an advantage? It could be used to set up a competition between the different machines for the best quality of information extraction from the data stream. Assuming that all machines receive virtually the same input (a transient stream they

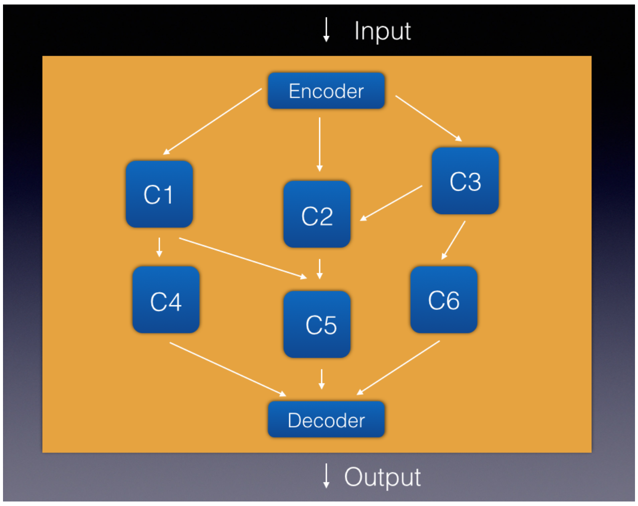

Figure 2: *An example of a soft computing machine's internal organization: A number of modules (C1, C2, etc.) are connected, each extracting other (possibly contradictory information from the encoded data stream. Integration and disambiguation must be sought to present the outside world with a defined reaction to the stream. The general character of processing is feed-forward from input to output, but occasionally there might also be feedback connections, mostly within modules (not shown here). Encoding is such that (i) it does not disturb the data stream; (ii) it can be discarded within the system. There is continuous input and output to the machine. Some of the components might be memory devices that store some high-level representation of extracted information.*

copy), they have the same starting conditions for information processing. Assuming that there are different routings of information, and perhaps different ways of encoding information within a machine, the results of information extraction and ultimately the reaction of the machines to the data stream might well be diverse. In such a situation, competition (and perhaps an evolutionary process) could set in for improving coding, information routing and processing to allow for the extraction of the optimal amount of high-level information or "knowledge," as it might be legitimately called.

Bibliography

[1] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight Jr, R. Nagpal, E. Rauch, G. Sussman, and R. Weiss. Amorphous Computing. *Communcations of the ACM*, 43(5):74–82, 2000.

[2] C. Bennett. Cosmology from start to finish. *Nature*, 440:1126 – 1131, 2006.

[3] M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, Y. Xing, and S. Zdonik. Scalable distributed stream processing. In *Proc. Conf. on Innovative Data Syst. Res*, 2003.

[4] E. Codd. *Cellular automata.* Academic Press, 1968.

[5] G. Dodig-Crnkovic. Significance of models of computation, from turing model to natural computation. *Mind and Machines*, 21:301–322, 2011.

[6] G. Dodig-Crnkovic. Info-computationalism and morphological computing of informational structure. In P. L. Simeonov, L. S. Smith, and A. C. Ehresmann, editors, *Integral Biomathics*, pages 97–104. Springer, Berlin Heidelberg, 2012. ISBN 978-3-642-28111-2.

[7] G. Dodig-Crnkovic. The info-computational nature of morphological computing. In V. C. Mueller, editor, *Philosophy and Theory of Artificial Intelligence*, volume 5 of *Studies in Applied Philosophy, Epistemology and Rational Ethics*, pages 59–68. Springer, Berlin Heidelberg, 2013. ISBN 978-3-642-31674-6.

[8] F. Ferrari, J. Nielsen, P. Questa, and G. Sandini. Space variant imaging. *Sensor Review*, 15:18–20, 1995.

[9] E. Fredkin. Finite nature. In *Proceedings of the XXVIIth Rencotre de Moriond*, 1992.

[10] L. Golab and M. T. Özsu. Issues in data stream management. *SIGMOD Rec.*, 32 (2):5–14, June 2003. ISSN 0163-5808.

[11] H. Hauser, A. Ijspeert, R. Füchslin, R. Pfeifer, and W. Maass. Towards a theoretical foundation for morphological computation with compliant bodies. *Biological cybernetics*, 105:355–370, 2011.

[12] A. v. Humboldt. *Kosmos, Entwurf einer physischen Weltbeschreibung*. Cotta, Stuttgart, 1845. (*Cosmos: A sketch of a physical description of the universe*, translated by E.C.Otté, London, Henry G. Bohn, 1848).

[13] K. Mainzer and L. Chua. *The Universe as Automaton - From Simplicity and Symmetry to Complexity*. Springer, Berlin Heidelberg, 2012.

[14] R. Pfeifer, F. Iida, and G. Gómez. Morphological computation for adaptive behavior and cognition. *International Congress Series*, 1291:22–29, 2006.

[15] K. Zuse. *Rechnender Raum*. Vieweg und Sohn, Braunschweig, 1969. (*Calculating Space*, MIT Technical Translation AZT-70-164-GEMIT, Massachusetts Institute of Technology, Cambridge, MA, 1970).