EMBODIED COMPUTATION

HEIKO HAMANN* and HEINZ WÖRN
Institute for Process Control and Robotics, Universität Karlsruhe (TH)
Engler-Bunte-Ring 8, 76131 Karlsruhe, Germany

Received May 2007
Revised August 2007
Communicated by Guest Editor

ABSTRACT
The traditional computational devices and models, such as the von Neumann architecture or the Turing machine, are strongly influenced by concepts of central control and perfection. The standard models of computation seem to cover the reality of computation only partially and lack, in particular, in the ability to describe more natural forms of computation. In this paper we propose the concept of embodied computation, a straight forward advancement of well known concepts such as amorphous computing, emergent phenomena and embodied cognitive science. Many embodied microscopic computational devices form a single macroscopic device of embodied computation. The solution to computational problems emerges from a huge amount of local interactions. The system's memory is the sum of the positional information and possibly of the internal states. Such systems are very robust and allow different methodologies to analyze computation. To back this theoretic concept some results based on simulations are given and potential benefits of this approach are discussed.

Keywords: unconventional computation, swarm intelligence, swarm robotics, natural computation, embodied computation

1. Introduction

The history of computer science and the engineering of computational devices is very impressive given the short period of time since this field exists. The provided computational power is applied to a huge variety of fields to predict or to control complex systems, to process huge amounts of data, and to make everyday items "intelligent". At the same time theoretical computer science has led us to a better understanding of computation itself. Lower and rather upper bounds of complexity help software engineers to optimize algorithms and not to try to find impossible solutions.

---

*hamann@ira.uka.de

However, besides this success there are unsolved problems that proved to be very hard over the years. This includes the slow advancements in artificial intelligence and unanswered fundamental questions in complexity theory.

Since about two decades alternative approaches were developed including some being not started explicitly in the context of computation but having now an impact to this field as well. Examples are emergent computation, cellular automata, chaos theory, or the so-called computation at the edge of chaos as well as approaches in biology and physics [8,17,23,9,32]. However, these approaches are of little or no help to the theory of computation so far. This is on the one hand due to the lack of an alternative theory and might be, on the other hand, a problem of unifying helpful approaches to one in order to explain computation.

The main purpose of this paper is to propose a new way of thinking about computation that exists already in many parts within many fields but not explicitly and unified. The authors believe in the utility of such an alternative theory of computation, following Marvin Minsky: "Unless you understand something in several different ways, you are likely to get stuck" [26]. At first, we give a short overview of the history of traditional computational devices. Then we present the concept of embodied computation in section . In section we give the results of two simulated scenarios, in section the proposed concept is compared to related work, and in section it is discussed.

2. A Short History of Computational Devices

The origin of our computational models and devices of today and the first actual ideas on the issue of how computation works, or what it is, and how it could be automated, goes back to Wilhelm Schickard in the year 1623 [16]. Here, we focus on the later but more sophisticated approach by Charles Babbage in 1821 [6]. As all ideas, also this one was, of course, influenced by the current zeitgeist. It is the time of the industrial revolution and the location is its origin – Great Britain. The rapid changes led to exaggerated expectations, such as the belief almost everything would be solvable by steam engines; and this is presumably true for Babbage, too. He wanted to improve the time-consuming and erroneous process of calculating mathematical tables manually. His idea was to build a machine that performs calculations mechanically with high reliability. He wanted to build a machine of perfection. The strictly mathematical and perfectionist background of the idea is also indicated by Ada Lovelace: "The Analytic Engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves" [6]. Babbage also introduced the strict separation of memory and CPU, what he called "store" and "mill" [6].

Later approaches to building computational devices were also based on the thought of a complex machine with wheels and gears. Alan Turing at Bletchley Park was clearly influenced by this idea, too. This is reflected in both his computational devices and his computational model – the Turing machine. The proposal of the first computer in a form as we know it today, the Mark I, included a detailed description of Babbage's work and, thus, shows the influence by the idea of mechanical calculation [6]. Although the devices switched over time from mechanical, over electromechanical to electronic principles, the underlying fundamental idea was, and still is, in a tradition of Babbage.

Summarizing it can be said: The traditional computing devices solve, by the majority, analytical tasks in a reliable and repeatable way. They use sophisticated algorithms being designed to be executed step by step. The hardware is fragile, in the sense of an existing single point of failure, and there is a strict separation between memory and CPU.

3. The Concept of Embodied Computation

The meaning of computation according to WordNet is "determining something by mathematical or logical methods" or "problem solving that involves numbers or quantities" [34]. However, our concept denies partly the involvement of strict logic and numbers. We keep the term computation because it has de facto a broader meaning and absorbs almost every process that converges to a fixed point over time, although some might prefer the term information processing. By following our concept, computation is originally not bounded to analytical approaches and tasks but denotes the process of solving a problem. We state that the natural approach to computation is essentially different from the computational devices of the last 200 years.

The term computation in a very broad sense does not exclude our concept of embodied computation but it also does not include it explicitly. We define the concept of embodied computation:

Device of Embodied Computation – *A device performing embodied computation satisfies the following requirements:*

- *It consists of at least two levels with a micro-macro relationship, i.e. a large number of embodied, possibly mobile, microscopic computational devices (MCD) of limited abilities form a single macroscopic computational device. This might be extended to a cascade of many levels.*
- *MCD are only loosely coupled among each other by means of short range communication or stigmergy without a fixed topology.*
- *MCD are sensor, actuator, CPU, and memory in one entity.*
- *The system is highly adaptive (often also called self-organized).*
- *As a consequence of the above characteristics the solution to the macroscopically defined task emerges from the microscopic level and the whole system is extremely robust, i.e. even a high percentage of damaged MCD has no influence to the effectiveness of the whole system.*

We are using "emergence" only as a shortcut to the accordingly associated properties of such systems, which are otherwise hard to be communicated. Definitions and the theory of emergence have proven to be of little use or being even misleading and destructive. A discussion of this problem would be beyond the scope of this paper.

A device as defined above is able to solve problems that cannot be solved by one of its parts alone. The solution to the problem emerges due to the high number of local interactions between the MCD. An MCD could be, for example, a robot of limited abilities, a biological cell, or a molecule. The main differences of embodied computational devices compared to traditional concepts are:

- Memory and CPU are not separated.
- Due to the embodiment the physical world is a relevant part of computation, e.g. positional information.
- There is no globally defined algorithm solving the task step by step.
- The solution is not repeatable and could be suboptimal or even wrong.
- The system is very robust.

We see two main benefits in investigating such computational systems: First, this kind of system might have the highest accomplishable robustness of all computational systems because it shows a maximum of local organization. Second, due to the involvement of embodiment and spatiality this might increase the possibilities of investigating and analyzing computation, for example, by applying methods of statistical mechanics or simple geometry.

An example of an embodied system is a leaf with its stomatal apertures [23]. The optimization problem needed to be solved by a leaf is to maximize CO_2 uptake and minimize water loss. This is done by the stomata interacting with neighboring stomata forming synchronized patches. Other examples are foraging ants [4], a crowd of people trying to leave a room [12], or a robotic swarm performing collective perception [27,29].

4. Results of Simulations

4.1. Steiner Tree Problem

The Euclidean Steiner Tree problem is defined as follows: A given set of points has to be connected by lines of minimal length and, in contrast to the minimal spanning tree problem, it is allowed to add extra points, called *Steiner points*. There are many applications to this problem in circuit design, mining, and network design. A lot of work has been done to solve this problem [14,25,5].

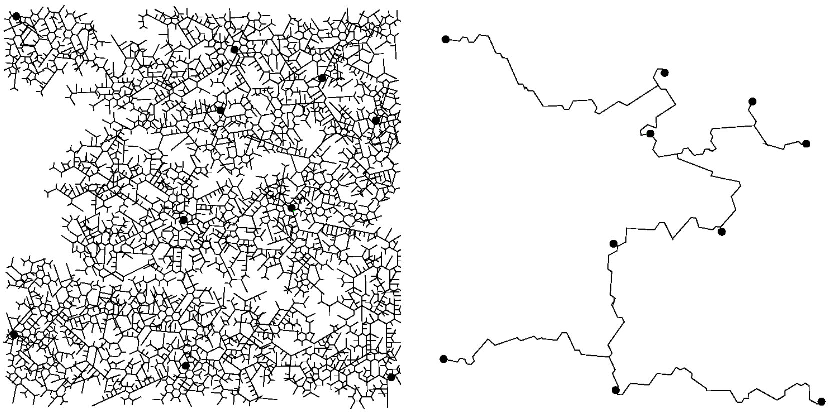
Here, we propose a way of approximating a solution to this problem with the principles of embodied computation. We assume objects being placed in a bounded plane called *seeds*. Each seed represents a point of the Steiner Tree problem to be solved. The MCD are a large group of mobile robots moving on that plane. They are equipped with sensors allowing them to perceive other robots and the seeds within a very bounded area compared to the dimensions of the plane. If a robot finds a yet uncovered seed it stops there and communicates to passing robots its internal state – the seed covering state. Another robot approaching this covering robot stops as well after having maximized the angle between itself, the covering robot, and the seed, i.e. forming a straight line. We say that these robots are connected. Up to three connections per robot are allowed. If a third robot approaches another one being already connected to two, it will connect. The angles between all four robots will be maximized again, i.e. forming angles of 120 degrees, but only if this does not require the movement of more than two robots. The angle of 120 degrees is not arbitrarily chosen, as the optimal solution of a Steiner Tree problem always consists of Steiner points of degree three and angles of 120 degrees only. Additionally, we want to cover as much space in the plane as possible with a minimal number of stopped robots. The optimal solution to this tiling problem is provably the hexagon as it is found in honey-combs [11]. However, the robots form only partial hexagons because the result should be trees.

Using this control algorithm the robots perform a process similar to diffusion-limited aggregation [31]. Hopefully, at each seed a tree will grow. At some point in time, provided that there are enough robots, two trees will be connected, later more will join. When only one huge tree is left, cf. Fig. 1(a), or at an assigned time, a new process is started: All robots being connected to only one other robot, i.e. they are a leaf, will cut this connection and leave. In a chain reaction all unnecessary robots cut their connections and a tree consisting of a relatively small number of robots is left, cf. Fig. 1(b). Robots being connected to three other robots after this reduction represent Steiner points. By straightening the connections between the seeds and the Steiner points, i.e. releasing robots in between, we get the approximated Steiner Tree, cf. Fig. 1(c). Note, the positions of the Steiner points were determined dynamically during the tree growth process. Interpreting this in the sense of swarm intelligence the Steiner points are the result of a collective decision that emerges from the numerous agent-agent interactions.

This heuristic is very robust to breakdowns of single MCD although it might seem very inefficient. Additionally, it is scalable because of its totally local approach. Whether this method can be a fast way of approximating a Steiner Tree is a question of the reaction times and speed of the robots. At the time mass production of such devices will become possible, this scenario might be feasible [29]. However, the relevance of this is meant to be of academic use only.

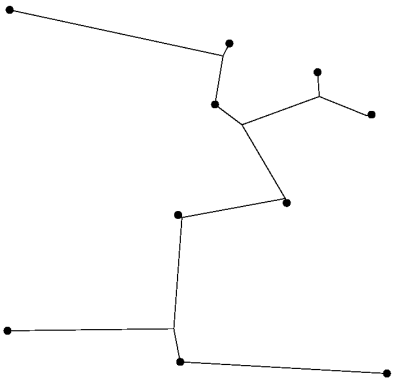
4.2. Density Classification Task

The density classification task is an example of an emergent and distributed computational problem [22]. The input is an array of length N with red and green elements. The problem to solve is: Are there more red or more green elements? For a standard computer this task would be simplistic but it was designed for cellular automata. They perform a very different type of computation that does not seem to be connected to our sense of logic. Here, we introduce an embodied computation variant of this problem. Again we have a bounded plane in which we put, randomly distributed, N robots of two states, red and green, in the given ratio. Each of them performs a collision avoiding behavior, i.e. if another robot gets too close (entering a circular area defined by the avoidance distance) it will turn around and try to increase this distance. At each avoided collision the robot remembers the color of the other one. After five collisions it changes its state to the color it encountered at most. Using this simple behavior we hope for convergence of the whole group of robots to one state. Then we interpret this state as the answer to the above problem. Unquestioned, we will not always get the right answer. Especially, this is true for the balanced case: the initial state of 50% red and 50% green robots. But also for values close to this ratio, initial fluctuations will lead to wrong decisions.



(a) A late stage of the network growth process.

(b) The reduced tree.



(c) The final approximated Steiner Tree.

Figure 1: An embodied computation approach to approximately solve the Steiner Tree problem.

Unexpectedly, in our simulation we encountered an error rate of only 18% for ratios of 0.5 ± 0.05 independent of N, for $N > 100$.

We analyzed the length of time till convergence depending on N for an initial ratio of 0.5 ± 0.05. Beginning with a small N this time decreases exponentially with increasing N and a fixed area, i.e. the problem becomes easier with increasing problem size. However, at the same time, we are also increasing the computational power of the system. Then, at some critical value of N, there is no further decrease in the length of time. This happens if the area needed by the robots (defined by a circle with the radius of the avoidance distance) covers about 19% of the plane. For even bigger values of N the runtime increases. In Fig. 2(a) the relationship between time till convergence, the state ratio, and the number of robots is given. As expected, the maximal time is needed for ratios close to 0.5 and decreases for either a majority of red or green robots. Remarkably, the decrease in time by increasing N does not seem to be smooth, as a jump between $N = 81$ and $N = 94$ indicates (corresponding to an area need in the above sense of 2.5% and 3.0%, respectively).

In Fig. 2(b) an irregularity is indicated as well. It shows the change of the state ratio per time step depending on the state ratio for varied N. Beginning with small N the change per step increases independently of the state ratio. However, for bigger values of N the region around 0.5 falls behind till there is no increase anymore and the ratio change per time step remains at about 0.0013.

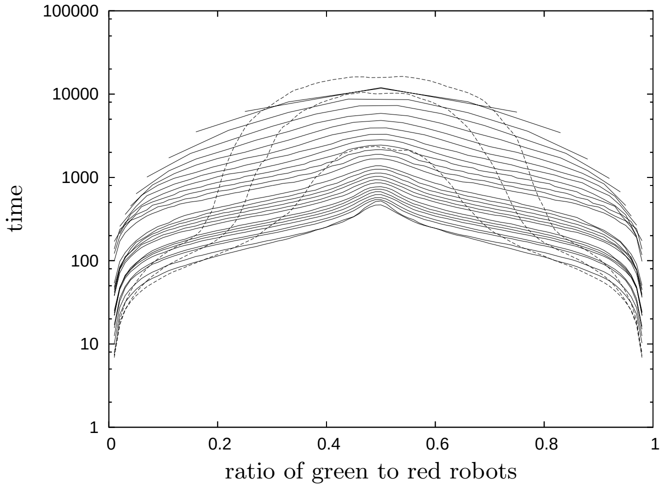
With these investigations we wanted to show that embodied computation is less intuitive than traditional computation.

5. Related Work and Potential Benefits

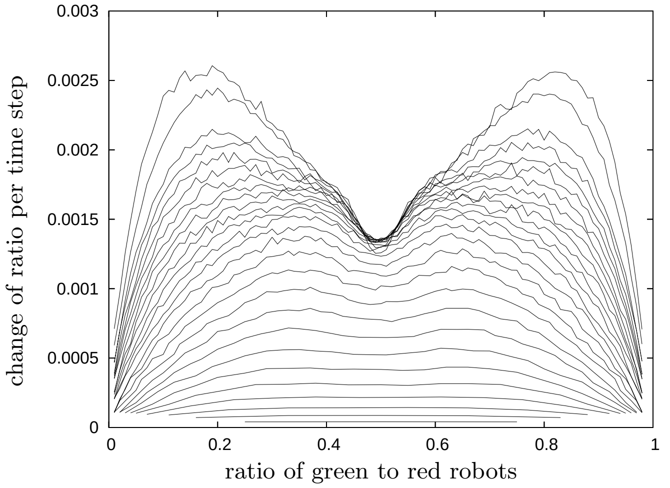
The idea of embodied computation is strongly related to the concept of amorphous computing [1]. Similarities are the assumption of high agent numbers, the abandonment of perfection in favor of a stochastic concept of computation, the restriction to local interactions only, and the use of metaphors from biology. A very recent comment by Sussman can be seen in this context, as well, addressing the problem of engineering robust systems: "It is hard to build robust systems: systems that have acceptable behavior over a larger class of situations than was anticipated by their designers" [30]. The research of amorphous computing is focused on the discrete approximation to a continuous "smart" medium and mostly on immobile particles. The emphases of embodied computation, on the other hand, being formed from a background of swarm robotics and swarm intelligence are the computational theoretic aspects as well as the concepts of mobility and emergence.

In another very recent work the authors propose a heuristic for multi-robot rendezvous based on insights that are essentials of embodied computation: "The key insight that underlies our methods is that the physical locations of the robots themselves could be considered as an approximate solution to the entire problem. An individual robot can move itself, thus refining the current solution approximation. No representation of the problem, or the current solution, needs to be held by any robot: they manifest the solution by their physical configuration" [18].

Embodied computation is also connected to the concept of embodied cognitive science as it is postulated by Pfeifer [24]. This research aims primarily at creating



(a) Average time till convergence over the state ratio with varying the number of robots; solid lines from the top downwards: $N \in \{4, 6, 9, 13, ..., 328, 400, 500\}$; dashed lines from the bottom upwards: $N \in \{700, 1000, 1400\}$.



(b) Average change of state ratio per time step with varying the number of robots, from the bottom upwards: $N \in \{4, 6, 9, 13, ..., 328, 400, 500\}$.

Figure 2: Analysis of the time till convergence and the change in the state ratio per time step depending on the state ratio and the number of robots N.

artificial intelligence and claims that it is only possible by a holistic approach unifying body and mind. The connection to embodied computation is the approach of cheap design, i.e. designing, for example, a robot in a way simplifying the control mechanism needed to make it walk [15]. This can be viewed as a decentral system of components forming together a leg with the components interacting and, thus, computing the dynamics of walking on their own. This is a form of emergence because the single parts have no concept of walking.

A strong influence to the authors is the field of swarm intelligence and swarm robotics. Instead of viewing a robotic swarm as a "massive parallel computational system" [3], in the sense of putting together a huge amount of micro-controllers, we view an individual in an artificial swarm as the smallest computational unit of a computing system. This distinction seems to be small but the change of methodologies is bigger. The swarm is not only regarded as a distributed system but as a system consisting of two levels: The individual robots on the microscopic level and the holistic view of the swarm as a macroscopic computational entity.

Another related work is the concept of self-awareness in decentral systems as proposed by Mitchell [19,20]. The four principles of self-awareness in decentral systems are all strongly connected to the concept of embodied computation. Global information is encoded in patterns formed by MCD, randomness and parallelism are essential parts, and there is an intense interplay of bottom-up and top-down processes.

Other related fields are emergent, evolutionary, and natural computation including the ideas of evolving computational systems and using biological systems or cellular automata to build computational devices [8,2]. Three aspects of embodied computation being only little or not considered in computation theory are mentioned by Crutchfield: "A second aspect computation theory has dealt with little, if at all, is measuring structure in stochastic processes. [...] A third aspect computation theory has considered very little is measuring structure in processes that are extended in space. A fourth aspect it has not dealt with traditionally is measuring structure in continuous-state processes" [7]. He proposes an approach of probabilistic and spatial computation capable of analyzing a series of measurements of a continuous-state system. Although this is conceptually close to embodied computation it is too focused on serving as a tool to investigate emergence to be of more relevance here. Also philosophically interesting but of limited relevance here is the concept of "Calculating Space" trying to develop a digital model of physics [36]. The same is true for the similar work by Fredkin and Wolfram [10,33].

Special fields, such as population dynamics or to some extent even game theory, could be regarded as parts of computer science because they investigate computational systems. Finally, the fields of chaos theory and phase transitions are also related offering the possibility of sudden changes in the complexity of a computational process [17,21,7].

The authors see a clear benefit in the proposed approach to establish new ways of analyzing computation. This might be achieved by approved methods of statistical physics [35,28]. By interpreting the MCD as microscopic particles, equations describing Brownian motion with drift can be applied. Then the micro-macro link (determining the macroscopic behavior out of the microscopic behavior and vice versa) is solved by the methodology of physics. Especially, the analysis of the average behavior of a computational system, which is a hard task using traditional computer science, might be simplified.

6. Discussion

Why should we investigate alternative computational systems such as the proposed concept of embodied computation? The traditional computational models such as the Turing machine or the von Neumann architecture seem to cover only a small part of possible forms of computation. Although these models have helped immensely from the past until today to build fast computers, to analyze, and design sophisticated algorithms, there might be complete facets yet uncovered because we have focused too much on similar models.

For example, consider the restriction of separating memory and CPU. A typical characteristic of natural, computational entities is the unity of all components: "In the case of leaves, stomata are simultaneously the sensors of external information, the processing units that calculate how gas exchange regulation should occur, *and* the mechanisms for executing the regulation" [23]. Restating this, MCD are sensor, computing unit, memory, and actuator in one.

Additionally, the memory of an embodied computational system consists not only of the sum of the distributed memories but of three parts: First, there is the internal memory of the MCD if they have some at all. Second, the environment serves as a memory. For example, this can be done by stigmergy: A robot changes the environment locally. Later another robot senses this alteration and reacts accordingly to this information. Third, the positions of the mobile MCD can be sensed by others and can serve as memory. This was the main aspect of section presenting the Steiner tree example.

The essential implication of embodiment is that many interactions between the agents are not executed following an artificial protocol but the natural laws of physics. Thus, both the parameters and the variables are determined and "administrated" by the physical world. In contrast to the explicit representation of numbers hold in an electronic memory, being encoded, sensitive to slight errors, and treated by programs according to mathematical laws.

There is no global algorithm but it rather emerges from the local algorithms defining only the behavior of the MCD. The global behavior that actually defines the computation is only implicitly determined. An embodied computational system does not calculate like adding and subtracting numbers and it does not solve problems exactly.

Regarding robustness, it seems to be hardly possible, in general, to have a maximum amount of robustness within a single computational system and assuring to find the exact solution at the same time. In fact, this seems to be a trade-off as the common trade-off between exploration and exploitation [13]. If the system computes an exact solution then there is a certain task assigned to each component creating the possibility of a single point of failure. However, this consideration is only true in regard of the robustness of a single macroscopic system. There is a difference in case of redundancy on the macroscopic level: Only if we have multiple macroscopic computational systems we can have robustness and exactitude at the same time. On this higher level, there is no difference in the robustness between a bunch of personal computers and a bunch of embodied computational systems.

This approach centered on swarm robotics could also be substituted by methods of nano-technology. For example, the control algorithm of the robots as described in section could also be realized by special material with such characteristics and behavior to solve the Steiner Tree problem, i.e. particles that behave, in principle, as in natural diffusion-limited aggregation.

7. Summary and Outlook

In this paper we have presented a short history of computational devices relying heavily on the concepts of central control and perfection. Embodied computation as a construct of ideas was argued. The results of two simulations, related fields, and potential benefits were discussed. Neither the traditional nor the new concept proposed in this paper is generally superior to the other. Always both of them should be considered, be it in the context of an application or the struggle for insight.

In future work this concept should be formalized, the huge variety of concepts should be reviewed, and unified to form a sound theory of computation either as an alternative to traditional computation theory or combined with it.

Acknowledgments

Hamann is supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 Self-organizing Sensor-Actuator Networks.

References

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

[2] D. H. Ballard. *An Introduction to Natural Computation.* MIT Press, January 1999.

[3] G. Beni. From swarm intelligence to swarm robotics. In E. Şahin and W. M. Spears, editors, *Swarm Robotics – SAB 2004 International Workshop*, LNCS, pages 1–9, Santa Monica, CA, July 2005.

[4] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford Univ. Press, 1999.

[5] M. Chlebík and J. Chlebíková. Approximation hardness of the Steiner Tree problem on graphs. In *Proc. 8th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 2368 of *LNCS*, pages 170–179, Berlin, Germany, 2002. Springer-Verlag.

[6] B. Collier and J. MacLachlan. *Charles Babbage: And the Engines of Perfection.* Oxford University Press, November 1998.

[7] J. Crutchfield. The calculi of emergence: Computation, dynamics, and induction. *Pyhsica D*, 75(1-3):11–54, August 1994.

[8] J. P. Crutchfield and M. Mitchell. The evolution of emergent computation. *Proceedings of the National Academy of Sciences*, 92:10742–10746, November 1995.

[9] R. P. Feynman, T. Hey, R. W. Allen, and T. Hey. *Feynman Lectures on Computation.* Perseus Books Group, Cambridge, MA, USA, 2000.

[10] E. Fredkin. Digital mechanics: An informational process based on reversible universal

cellular automata. *Physica D*, 45:254–270, 1990.

[11] T. C. Hales. The honeycomb conjecture. *Discrete and Computational Geometry*, 25(1):1–22, 2001.

[12] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, September 2000.

[13] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, USA, 1992.

[14] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. North-Holland, 1992.

[15] F. Iida. Cheap design approach to adaptive behavior: Walking and sensing through body dynamics. In *International Symposium on Adaptive Motion of Animals and Machines*, 2005.

[16] F. W. Kistermann. Abridged multiplication – the architecture of Wilhelm Schickard's calculating machine of 1623. *Vistas in Astronomy*, 28(1):347–353, 1985.

[17] C. G. Langton. Computation at the edge of chaos: phase transitions and emergent computation. In *Proceedings of the ninth annual international conference of the Center for Nonlinear Studies on Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks*, pages 12–37, 1990.

[18] Y. Litus, P. Zebrowski, and R. Vaughan. Energy-efficient multi-robot rendezvous: Parallel solutions by embodied approximation. In *Workshop on Algorithmic equivalencies between biological and robotic swarms, Atlanta, USA*, June 2007.

[19] M. Mitchell. Self-awareness and control in decentralized systems. In *Working Papers of the AAAI 2005 Spring Symposium on Metacognition in Computation*. AAAI Press, 2005.

[20] M. Mitchell. Complex systems: Network thinking. *Artificial Intelligence*, 170(18):1194–1212, 2006.

[21] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7(2):89–130, 1993.

[22] N. H. Packard. Adaptation toward the edge of chaos. In J. Kelso, A. Mandell, and M. Shlesinger, editors, *Dynamic Patterns in Complex Systems*, pages 293–301. World Scientific, 1988.

[23] D. Peak, J. D. West, S. M. Messinger, and K. A. Mott. Evidence for complex, collective dynamics and emergent, distributed computation in plants. *Proceedings of the National Academy of Science*, 101(4):918–922, 2004.

[24] R. Pfeifer and J. C. Bongard. *How the body shapes the way we think – A new view of intelligence*. MIT Press, November 2006.

[25] G. Robins and A. Zelikovsky. Improved Steiner Tree approximation in graphs. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.

[26] W. Roush. Marvin minsky on common sense and computers that emote. Technology Review, July 2006.

[27] T. Schmickl, C. Möslinger, and K. Crailsheim. Collective perception in a robot swarm. In E. Şahin, W. M. Spears, and A. F. Winfield, editors, *Swarm Robotics – Second SAB 2006 International Workshop*, volume 4433 of *LNCS*, 2007.

[28] F. Schweitzer. *Brownian Agents and Active Particles. On the Emergence of Complex Behavior in the Natural and Social Sciences*. Springer-Verlag, Berlin, Germany, 2003.

[29] J. Seyfried, M. Szymanski, N. Bender, R. Estaña, M. Thiel, and H. Wörn. The I-SWARM project: Intelligent small world autonomous robots for micro-manipulation. In E. Şahin and W. M. Spears, editors, *Swarm Robotics – SAB 2004 International Workshop*, pages 70–83, Berlin, Germany, 2005. Springer-Verlag.

[30] G. J. Sussman. Designing for applications unanticipated by the designer. Keynote

at the *First IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Boston, Mass., USA, July 2007.

[31] J. Tom A. Witten and L. M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys. Rev. Lett.*, 19:1400 – 1403, 1981.

[32] S. Wolfram. Undecidability and intractability in theoretical physics. *Phys. Rev. Lett.*, 54:735–738, 1985.

[33] S. Wolfram. *A New Kind of Science*. Wolfram Media, May 2002.

[34] WordNet 3.0, Princeton University, 2007.
http://wordnet.princeton.edu/perl/webwn?s=computation.

[35] H. Wörn, R. Estaña, H. Hamann, and M. Szymanski. From micro to nano and swarm robotics. In M. Xie and S. Dubowsky, editors, *Advances in Climbing and walking robots. Proceedings of 10th International Conference (CLAWAR'07), Singapore, July 16-18*, pages 15–24, July 2007.

[36] K. Zuse. *Rechnender Raum.* Friedrich Vieweg u. Sohn, 1969.