Reprinted from

Robotics and Autonomous Systems

Robotics and Autonomous Systems 19 (1996) 23–31

Evolutionary teleomorphology

Thomas C. Henderson *, Alexei A. Efros
Department of Computer Science, University of Utah, Salt Lake City, UT 84112, USA

ELSEVIER

ELSEVIER

Robotics and Autonomous Systems 19 (1996) 23–31

Robotics and Autonomous Systems

Evolutionary teleomorphology

Thomas C. Henderson *, Alexei A. Efros

Department of Computer Science, University of Utah, Salt Lake City, UT 84112, USA

Abstract

The physical layout of organs and neural structures in biological systems is important to their functioning, and is the result of evolutionary selection forces. We believe this is true even at the individual neuron level, and should be accounted for in any bio-based approach. In particular, when transmission delay is taken into account, the physical layout problem (PLP) of neural centers and individual neurons has a great impact on any computation they perform. We demonstrate on a simple example that: (1) performance can depend crucially on the physical layout of the computational nodes in a system, and (2) evolutionary schemes can be used to find near-optimal solutions to PLP.

Keywords: Evolution; Teleology; Structure; Analog; Continuous; Robot body; Physical layout; Signal delay

1. Introduction

A large literature exists on the study of bio-based computing systems, including neural networks [4,5], artificial neurons [1,7], and analog computing schemes[9]. The main issues generally concern the model of the individual neurons, and the connectivity structure of the neurons in a network. Rarely does anyone take into consideration the issue of signal transmission delay due to the physical layout and the length of connections between nodes.

We believe that the physical layout of biological information processing systems is not random, but is the product of an evolutionary bias which selects based on performance. For example, it is most likely no accident that the visual field (which falls on the left of the retinas) maps to the right side of the human brain. It makes sense that there is a direct physical relation between the location of processing in the body versus the 3D origin of the attention getting activity in the world.

2. A simple test model

In order to explore this notion of optimal physical layout, we have developed the following simple artificial amoeba (AA) shown in Fig. 1. The AA is square-shaped and exists in the plane; its motion is restricted to be along the x-axis. AA has sensors S_1 to S_4 which sense toxins in its environment. There are two propulsion units, P_1 on the left which can propel AA to the right, and P_2 which can propel AA to the left.

AA also has four processing nodes, N_1 to N_4, and N_i receives input from sensor S_i. The node connectivity in Fig. 1 aims to organize the computation of a control value on each propulsion unit while allowing a comparison of the toxin levels at the two ends of AA (top and bottom are compared independently). Moreover, two nodes are allocated in a top/bottom disposition in order to add redundancy to the control

* Corresponding author.

PII S0921-8890(96)00031-0

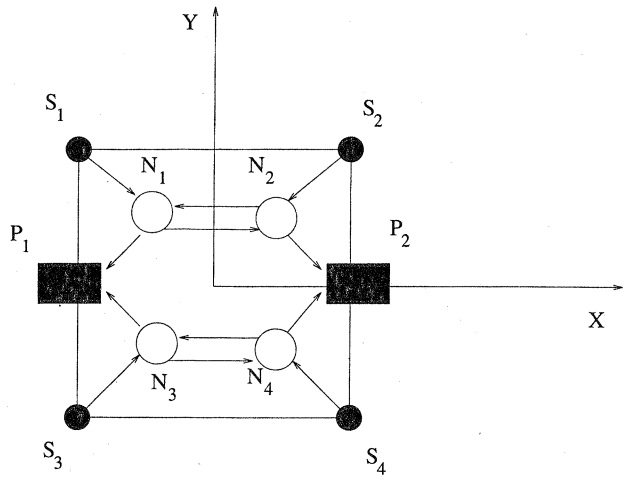


Fig. 1. Artificial amoeba.

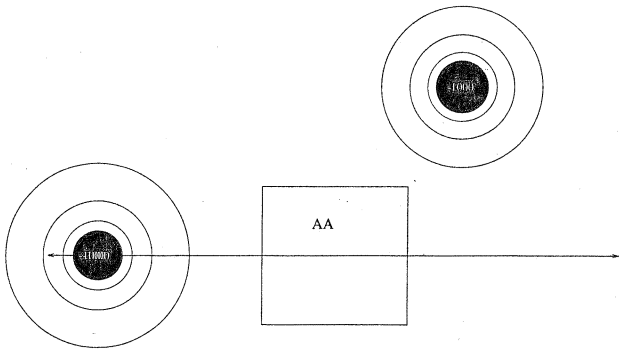


Fig. 2. Environment (bath) for AA.

of the actuators. Each node N_i has a location in the square body of AA, and this location determines the distances between sensors, nodes and actuators. The time required for a signal to travel along an arc is proportional to the length of the arc. Each arc can thus be viewed as a queue of values which propagate along the arc to the destination.

This AA model allows us to pose the following question concerning node layout: *What physical placement of the nodes N_1 to N_4 yields the best performing AA?*

Here we assume that the locations of the sensors and propulsion units are fixed. This is done to reduce the number of variables in the problem; moreover, allowing them to migrate would confuse the results of the processing node migration, but would not significantly alter the nature of the evolutionary process. We also need to give a more precise definition of *performance*.

The *life* of the AA is its time history when placed in a planar *bath* which includes one or more toxin sources (see Fig. 2). The toxin follows a $1/r^2$ law (i.e., the concentration of toxin falls off inversely proportional to r^2).

The performance of AA at the ith time step, $P^{(i)}$, is just the sum of the sensor values (a toxin has a negative value):

$$P^{(i)} = \sum_{j=1}^{4} S_j.$$

The overall performance of AA is the sum of the performances over all time steps:

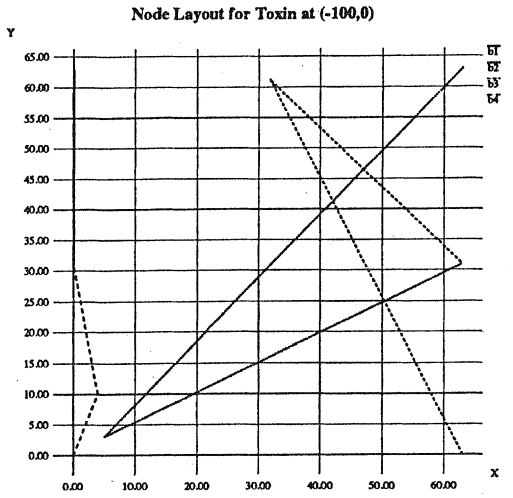


Fig. 3. Node layout for environment 1 – Toxin at $(-100, 0)$.

$$P(\mathrm{AA}) = \sum_{i=0}^{t} P^{(i)}.$$

Thus, the best performance is one which produces the greatest value for P; this corresponds to moving quickly away from the toxin source.

Before giving our solution to the PLP for this AA, we need to describe the internal computation of the AA. Each node N_i has its value initialized to zero, as does each arc. The simulation proceeds with the sensors relaying their values along the sensor-node arc. If the transmission rate along the arc is v units per time unit, then each arc is basically a fixed-length queue with $q = d/v$ values on it, and these move along the queue at one element per time unit. Each processing node N_i has two inputs, A and B, and outputs $\max(-(A + B), 0)$. The propulsion unit sums its inputs and if the result is a positive value, then it causes the AA to move that many units (if both propulsion units are activated, the net result is the difference of the two with resultant motion in the direction of the stronger push).

In terms of this model, we are looking for the best locations of nodes N_1 to N_4 so as to maximize the value of $P(\mathrm{AA})$. In order to determine the solution, we have used a genetic algorithm. First, a population of 200 random AAs are generated (i.e., 200 AAs with the nodes located in randomly generated positions). These are run independently for 200 time steps in a particular bath, and a standard genetic algorithm is used to produce the next generation (we use the GENESIS System [2,3,8] – in particular, see Grefenstette's manual for lots more references). The genetic string comprises the 4-node locations encoded as 6 bits for each coordinate:

$$b_{1,1}b_{1,2}\cdots b_{1,12}b_{2,1}b_{2,2}\cdots b_{2,12}\cdots b_{4,1}\cdots b_{4,12}$$

where $b_{i,1}$ to $b_{i,6}$ is the 6-bit x location and $b_{i,7}$ to $b_{i,12}$ is the 6-bit y location for N_i.

The genetic algorithms were run with parameters to achieve 20 000 total trials, a population size of 200, structure length of 48 bits, a crossover rate of 0.6 and a mutation rate of 0.001.

2.1. Environment 1: negative source at $(-100, 0)$

First we consider the case when AA is placed in a bath with a toxin source located at $(-100, 0)$ and with an intensity of $-100\,000$. Fig. 3 shows a typical resulting physical layout for the nodes (node N_1 lies on the y-axis). A histogram of the x-location values of the four nodes in the top performing AAs is given in Fig. 4, while a histogram of the y-location values is given in Fig. 5.

As can be seen from these graphs, when the toxin source is located to the left of the AA, then the nodes connected to the sensors on the left end up being placed as far left as possible (toward the y-axis) and

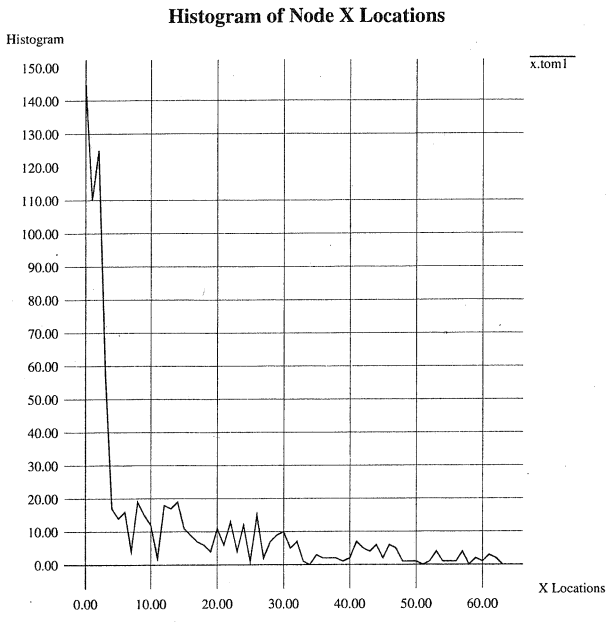


Fig. 4. Histogram of x-locations in top layouts.

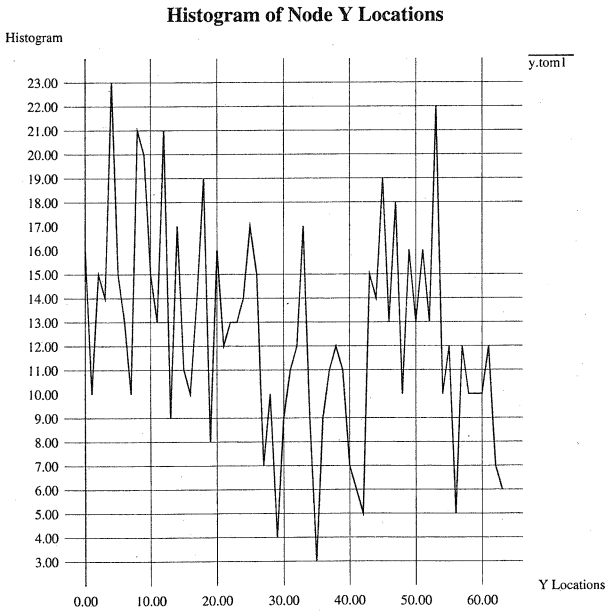


Fig. 5. Histogram of y-locations in top layouts.

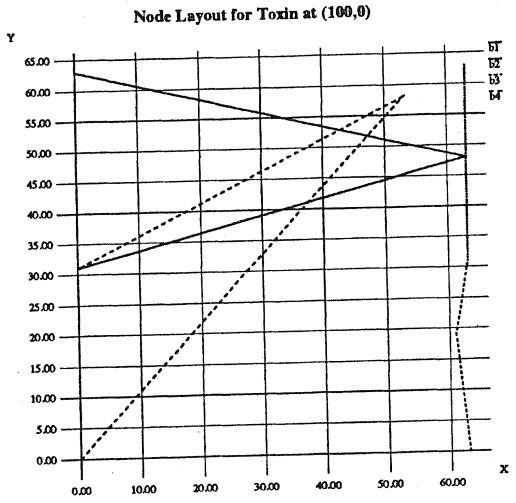


Fig. 6. Environment 2: −100 000 intensity toxin at (100,0).

between the sensor and the propulsion unit, while the nodes on the right move as far away as possible from the sensor to which they are connected. A ready explanation of this is that the resultant location of the left sensor handling nodes minimizes the time to starting a motion to the right (by firing the left propulsion units), while at the same time maximizing the time to the start of the right propulsion units (they cannot start until the signal travels the distance from the sensor to the processing node, and then back to the propulsion unit).

2.2. *Environment 2: negative source at (100,0)*

When the toxin source is located at (100,0) and with an intensity of −100 000, a symmetric result is obtained. A histogram of the x location values of the four nodes in the top performing AAs is given in Fig. 7, while a histogram of the y location values is given in Fig. 8. Fig. 6 shows a typical resulting physical layout for the nodes in this case.

As can be seen, the resulting locations of the processing nodes mirror those of Environment 1.

2.3. *Environment 3: negative sources at both (−100,0) and (100,0)*

The layout in this case has toxin sources of equal intensity on both sides of the AA. The performance scoring is done by running the AA with a left toxin source and a right toxin source each trial, and summing the two performances. The result in this case indicates that each side optimizes to respond to its sensors. Fig. 9 shows a typical resulting physical layout for the nodes in this case. Fig. 10 shows the x location histogram, while Fig. 11 shows the y location histogram for the top performing layouts.

3. Discussion and conclusions

This preliminary work supports the claim that physical layout plays a role in bio-based computing systems. We are currently looking into several more complicated scenarios, including the following sensors.

Positive and negative sensors: It is important to include positive reinforcement sensors, as well as avoidance-like sensors. The interaction of sensors responding positively, for example, to nutrients, also plays an important role in biological systems, and the final layout of processing nodes of both positive and negative feedback types requires study.

Activity sensors: Another aspect that we would like to explore is the use of cells within the organism which monitor the activity of other nodes and arcs, that is, *activity sensors*. These nodes monitor various sets of

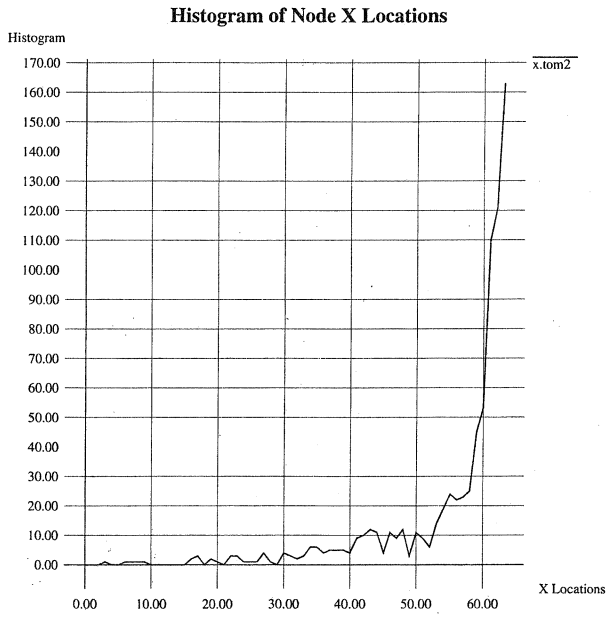


Fig. 7. Histogram of x-locations in top layouts.

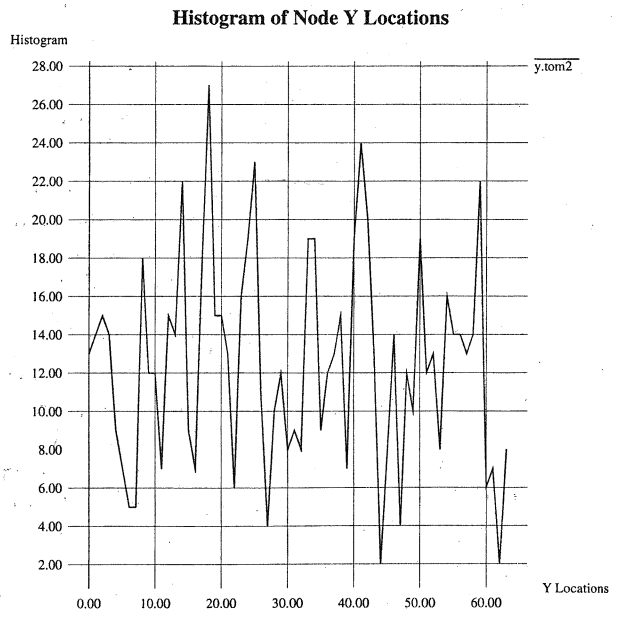


Fig. 8. Histogram of y-locations in top layouts.

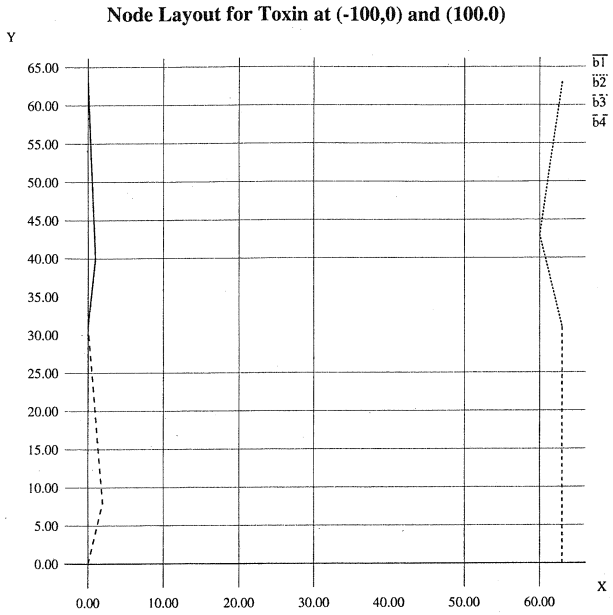


Fig. 9. Environment 3: $-100\,000$ intensity toxin at $(-100, 0)$ and (100,0).

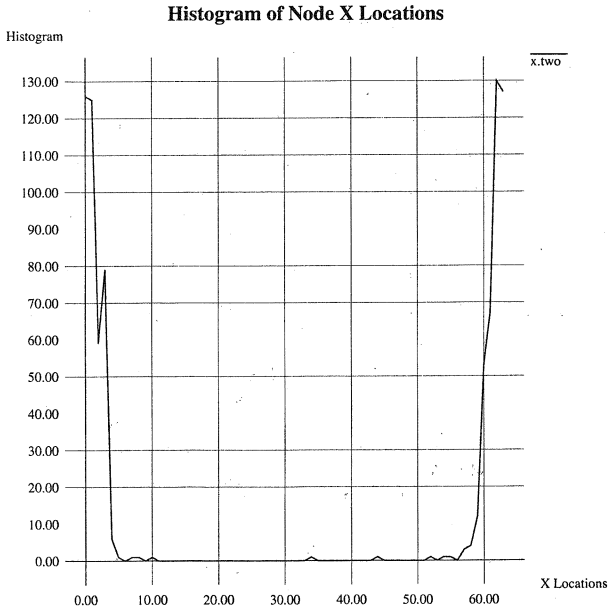


Fig. 10. Histogram of x-locations in top layouts.

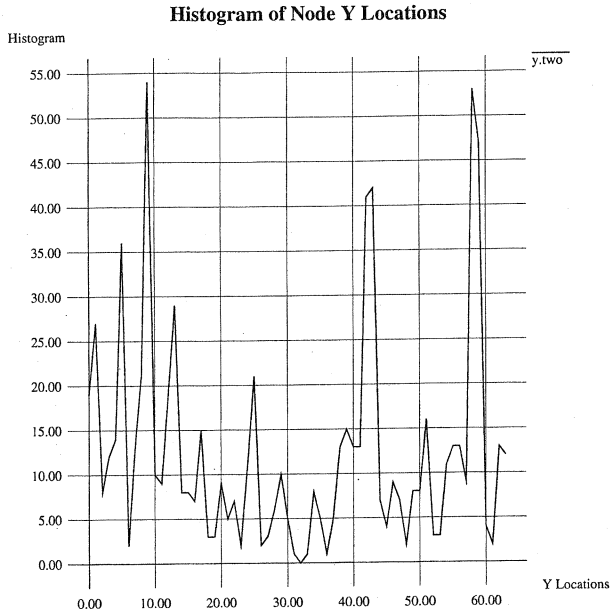


Fig. 11. Histogram of y-locations in top layouts.

nodes and arcs and respond to activity in that set. This permits the organism to respond directly to processing activity related directly to the 3D world origin of the stimulation, and this can occur before that stimulation has been completely analyzed. For example, neurons responding to activity in the right visual field might be monitored by such activity sensors and cause the head or body to turn in that direction before the visual information has been completely deciphered.

More realistic environments: Real environments do not present one simple negative or positive source to which all generations of the organism respond. It is essential to incorporate environments which have multiple sources, both positive and negative, as well as time-dependent variables, etc.

Physical prototypes: While simulation studies are of interest and provide insight into the nature of evolutionary teleomorphology, we intend to build analog mechanisms which have the capability of altering their physical layout in response to environmental forces. Thus, some form of *physical layout learning* should be supported by these artificial organisms. We believe that our work on artificial neurons provides one approach to this [6].

References

[1] D. Brettle and E. Niebur, Detailed parallel simulation of a biological neuronal network, *IEEE Computational Science and Engineering*, Winter (1994) 31–43.

[2] R. Collins and D. Jefferson, Antfarm: Towards simulated evolution, in C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen, eds., *Proc. Artificial Life II* (Addison-Wesley Reading, MA, 1990) 579–602.

[3] J.J. Grefenstette, A user's guide to genesis, Technical Report Version 5.0, 1990.

[4] S. Grossberg and M. Kuperstein, *Neural Dynamics of Adaptive Sensory-Motor Control* (North-Holland, Amsterdam, 1986).

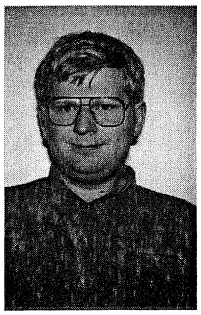
[5] R. Hecht-Nielsen, *Neurocomputing* (Addison-Wesley, Reading, MA, 1990).

[6] T.C. Henderson and A.A. Efros, Bio-based control for autonomous systems, in: Horst Bunke, Takeo Kanade and Hartmut Noltemeier, eds., *Modelling and Planning for Sensor-Based Intelligent Robot Systems* (World Scientific, Berlin, 1995) 431–442.

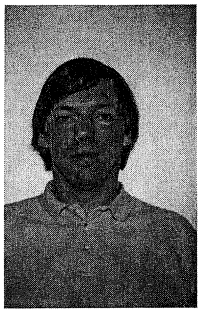
[7] F.C. Hoppensteadt, *An Introduction to the Mathematics of Neurons* (Cambridge University Press, Cambridge, 1986).

[8] D. Jefferson, R. Collins, C. Cooper, M. Byer, M. Flowers, R. Korf, C. Taylor, and A. Wang, Evolution as a theme in artificial life: The genesis tracker system, in: C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen, eds., *Proc. Artificial Life II* (Addison-Wesley, Reading, MA, 1990) 511–548.

[9] C. Mead, *Analog VLSI and Neural Systems* (Addison Wesley, Reading, MA, 1989).



Alexei A. Efros originally from St. Petersburg, Russia, is an undergraduate student in the Department of Computer Science at the University of Utah. He is a Barry Goldwater Scholar, and a Research Assistant in the computer vision group. His interests include cognitive science, artificial intelligence and computer vision. His major areas of current research are vision for mobile robots and genetic algorithms.



Thomas C. Henderson is with the Department of Computer Science at the University of Utah. He graduated from the University of Texas in 1979, after which he spent a year at INRIA in Roquencourt, France, and then a year at DFVLR in Oberpfaffenhofen, Germany. His professional interests include artificial intelligence, computer vision and robotics. Major areas of current research are robot behavior specification, multisensor integration and sensing strategies, and parallel vision algorithms.