From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

"CAM-BRAIN" ATR's BILLION NEURON ARTIFICIAL BRAIN PROJECT

Hugo de Garis

Brain Builder Group, Evolutionary Systems Department,
ATR Human Information Processing Research Laboratories,
2-2 Hikaridai, Seika-cho, Soraku-gun, Kansai Science City, Kyoto, 619-02, Japan.
tel. + 81 7749 5 1079, fax. + 81 7749 5 1008, degaris@hip.atr.co.jp

Abstract

This work reports on progress made in the first 3 years of ATR's "CAM-Brain" Project, which aims to use "evolutionary engineering" techniques to build/grow/evolve a RAM-and-cellular-automata based artificial brain consisting of thousands of interconnected neural network modules inside special hardware such as MIT's Cellular Automata Machine "CAM-8", or NTT's Content Addressable Memory System "CAM-CAM". The states of a billion (later a trillion) 3D cellular automata cells, and millions of cellular automata rules which govern their state changes, can be stored relatively cheaply in giga(tera)bytes of RAM. After 3 years work, the CA rules are almost ready. MIT's "CAM-8" (essentially a serial device) can update 200 million CA cells a second. It is likely that NTT's "CAM-CAM" (Cellular Automata on Content Addressable Memory) is essentially a massively parallel device, and will be able to update a *hundred billion* CA cells a second. Hence all the ingredients will soon be ready to create a revolutionary new technology which will allow thousands of evolved neural network modules to be assembled into artificial brains. This in turn will probably create not only a new research field, but hopefully a whole new industry, namely "brain building". Building artificial brains with a billion neurons is the aim of ATR's 8 year "CAM-Brain" research project, ending in 2001.

1. Introduction

ATR's CAM-Brain project resulted from the experience of the author's thesis work, in which he evolved neural net modules (using concatenated bit-string weights) to control the behavior of a simulated quadruped called "LIZZY", which could walk straight, turn left, turn right, peck at food and mate (de Garis 1994). Each of these behaviors was controlled by the time varying outputs of a single evolved neural network module, and applied to the angles of the leg components of LIZZY. (As far as he is aware, the author was the first person to evolve neural net dynamics (de Garis 1991), (in the form of walking stick-legs "Walker")). Switching between behaviors involved taking the outputs from one neural net module and feeding them into the inputs of the next module. The next step was to evolve neural net detectors, e.g. for frequency, signal strength, signal strength difference, etc. Finally, neural net "production rule" modules were evolved which could map conditional inputs from detectors to output behaviors. Thus an "intelligent" artificial creature was built, which could detect prey, mates and predators, and then approach and eat or mate, or turn away and flee.

Virtually every neural net that the author tried to evolve, evolved successfully. *The evolution of these fully connected neural network modules proved to be a very powerful technique.* This success made a deep impression on the author, reinforcing his dream of being able to build much more complex artificial nervous systems, even artificial brains. However, every time the author added a neural net module to the Lizzy simulation, its speed on the screen was slowed (on a Mac II computer). Gradually, the necessity dawned on the author that some kind of evolvable hardware solution (de Garis 1993) would be needed to evolve large numbers of neural net modules and at great speed (i.e. electronic speed) in special machines the author calls "Darwin Machines" [de Garis 1993]. Evolving artificial brains directly in hardware remains the ultimate future goal of the author, but in the meantime (since the field of evolvable hardware (EHW, E-Hard) is today only in its infancy), the author compromises by using cellular automata to grow/evolve neural nets in large numbers in RAM, which is cheap and plentiful. (It is now possible to have a gigabyte (a billion byes) of RAM in one's work-station). By using cellular automata based neural nets which grow and evolve in gigabytes of RAM, it should be possible to evolve large numbers (tens of thousands) of neural net modules, and then assemble them (or even evolve their interconnections) to build an artificial brain. The bottleneck is the speed of the processor which updates the CA cells. State of the art in such processors is MIT's "CAM-8" machine, which can update 200,000,000 CA cells a second.

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

Recently, it has been suggested by the author's ATR colleague Hemmi, that NTT's Content Addressable Memory System "CAM-CAM" (which should be ready by the end of 1997) might be able to update CA cells at a rate *thousands* of times faster than the MIT machine, i.e. at a *hundred billion* CA cells per second. NTT's machine is massively parallel. Hemmi and his programmer assistant Yoshikawa are now (December 1995) busily engaged in writing software to convert the author's CA rules (in 2D form) into Boolean expressions suitable for the NTT machine. If they succeed in applying this machine to CAM-Brain, then a new era of brain building can begin, because the ability to evolve thousands of neural net modules would become realistic and very practical (for example, to evolve a neural net module inside a cubic space of a million CA cells, i.e. 100 cells on a side, at a hundred billion cells a second, would take at most about 500 clock cycles, i.e. about five milliseconds. *So the evolution of a population of 100 chromosomes over 100 generations could all be done in about one minute.)* All the essential ingredients for brain building would be available (lots of RAM, the CA rules, and fast CA processors). Even if Hemmi does not succeed, then a new machine can be designed to be thousands of times faster than the CAM-8 machine. The author believes the CAM-Brain breakthrough is either less than a year away, or at most only a few years away (the time necessary to design and build a "Super-CAM" machine, probably with the help of NTT).

The above gives an overview of the CAM-Brain research project. What now follows is a more detailed description of CAM-Brain, showing how one grows and evolves CA based neural net modules in 2D and 3D. We begin with the essential idea. Imagine a 2D CA trail which is 3 cells wide (e.g. Fig. 2). Down the middle of the trail, send growth signals. When a growth signal hits the end of the trail, it makes the trail extend, or turn left, or right, or split etc., depending upon the nature of the signal (e.g. see Figs. 3-6). It was the author who hand coded the CA rules which make these extensions, turns, splits etc. happen. The CA rules themselves are *not* evolved. It is the *sequence* of these signals (fed continuously over time into an initialized short trail) that is evolved. This sequence of growth signals is the "chromosome" of a genetic algorithm, and it is this sequence that maps to a cellular automata network. When trails collide, they can form "synapses" (e.g. see Fig. 7). Once the CA network has been formed in the initial "growth phase", it is later used in a second "neural signaling phase". Neural signals move along CA-based axons and dendrites, and across synapses etc. The CA network is made to behave like a conventional artificial neural network (see Fig. 11). The outputs of some of the neurons of the complex recurrent networks which result can be used to control complex time dependent behaviors whose fitnesses can be measured. These fitness values can be used to drive the evolution. By growing/evolving thousands of neural net modules and their interconnections in an incremental evolutionary way, it will be possible to build artificial brains. According to the CAM developers at MIT, it is likely that the next generation of CAMs will achieve an increase in performance of the order of thousands, within 5 years. However, to be able to evolve a billion neuron artificial brain by 2001 (ATR's goal), it is likely that a "nano-CAM" machine (i.e. one which uses nano-scale electronic speeds and densities) will need to be developed. To this end, we are collaborating with an NTT researcher who has developed a nanoscale electronics device, who wants to combine huge numbers of them to behave as nano-scale cellular automata machines.

In the summer of 1994, a two dimensional CAM-Brain simulation was completed which required 11,000 hand crafted CA state transition rules. It was successfully applied to the evolution of maximizing the number of synapses, outputting an arbitrary constant neural signal value, outputting a sine wave of a desired arbitrary period and amplitude and to the evolution of a simple artificial retina which could output the vector velocity of a "white line" which "moved" across an array of "detector" neurons. Work on the 3D simulation should be completed by early 1996, and is expected to take about 150,000 hand crafted CA rules. The Brain Builder Group of ATR took possession of one of MIT's CAM8 machines in the fall of 1994. At the time of writing (December 1995) the porting of the 2D rules from a Sparc20 workstation to the CAM8 is nearing completion. If the porting of the rules of the 3D simulation to this machine is not possible, then a "SuperCAM" machine will be designed specifically for CAM-Brain, with the collaboration of the Evolutionary Technologies (ET) group of NTT, with whom our Brain Builder group of ATR's Evolutionary Systems (ES) group, collaborates closely. The complexity of CAM-Brain will make it largely undesignable, so a (directed) evolutionary approach called "evolutionary engineering" is being used. Neural networks based on cellular automata (Codd 1968), can be grown and evolved at electronic speeds inside state of the art cellular automata machines, e.g. MIT's "CAM8" machine, which can update 200 million cells per second (Toffoli & Margolus 1990). Since RAM is cheap, gigabytes of RAM can be used to store the states of the CA cells used to grow the neural networks. CA based neural net modules are evolved in a two phase process. Three cell wide CA trails are grown by sending a sequence of growth signals (extend, turn left, turn right, fork left, fork right, T fork) down the middle of the trail. When an instruction hits the end of the trail it executes its function. This sequence of growth instructions is treated as a chromosome in a Genetic Algorithm (Goldberg 1989) and is evolved. Once gigabytes of RAM and electronic evolutionary speeds can be used, genuine brain building, involving millions and later billions of artificial neurons, becomes realistic, and should become concrete within a year or two. The CAM-Brain Project should revolutionize

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

the fields of neural networks and artificial life, and in time help create a new specialty called "Brain Building", with its own conferences and journals.

This work consists of the following sections. Section 2 describes briefly the idea of "Evolutionary Engineering", of which the CAM-Brain Project is an example. Section 3 describes how neural networks can be based on cellular automata (Codd 1968), and evolved at electronic speeds. Section 4 presents some of the details of CAM-Brain's implementation. Section 5 shows how using cellular automata machines will enable millions of artificial neural circuits to be evolved to form an artificial brain. Section 6 discusses changes needed for the 3D version of CAM-Brain. Section 7 deals with recent work. Section 8 deals with future work and section 9 summarizes.

2. Evolutionary Engineering

Evolutionary Engineering is defined to be *"the art of using evolutionary algorithms (such as genetic algorithms (Goldberg 1989)) to build complex systems."* This work reports on the idea of evolving cellular automata based neural networks at electronic speeds inside cellular automata machines. This idea is a clear example of evolutionary engineering. Evolutionary engineering will be increasingly needed in the future as the number of components in systems grows to gargantuan levels. Today's nano-electronics for example, is researching single electron transistors (SETs) and quantum dots. Probably within a decade or so, humanity will have full blown nanotechnology (molecular scale engineering), which will produce systems with a trillion trillion components [Drexler 1992]. The potential complexities of such systems will be so huge, that designing them will become increasingly impossible. However, what is too complex to be humanly designable, might still be buildable, as this work will show. By using evolutionary techniques (i.e. evolutionary engineering), it is often still possible to *build* a complex system, even though one does not understand how it functions. This arises from the notion of the "complexity independence" of evolutionary algorithms, i.e. so long as the (scalar) fitness values which drive the evolution keep increasing, the internal complexity of the evolving system is irrelevant. This means that it is possible to successfully evolve systems which function as desired, but which are to complex to be designable. The author believes that this simple idea (i.e. the complexity independence of evolutionary algorithms) will form the basis of most 21st century technologies (dominated by nanotechnology (Drexler 1992)). Thus, evolutionary engineering can "extend the barrier of the buildable", but may not be good science, because its products tend to be black boxes. However, confronted with the complexity of trillion trillion component systems, evolutionary engineering may be the only viable method to build them.

3. Cellular Automata Based Neural Networks

Building an artificial brain containing billions of artificial neurons is probably too complex a task to be humanly designable. The author felt that brain building would be a suitable task for the application of evolutionary engineering techniques. The key ideas are the following. Use evolutionary techniques to evolve neural circuits in some electronic medium, so as to take advantage of electronic speeds. The medium chosen by the author was that of cellular automata (CA) (Codd 1968), using special machines, called "Cellular Automata Machines (CAMs)", which can update hundreds of millions of CA cells a second (Toffoli & Margolus 1990).

CAMs can be used to evolve the CA based neural networks at electronic speeds. The states of the cellular automata cells can be stored in RAM, which is cheap, so one can have gigabytes of RAM to store the states of billions of CA cells. This space is large enough to contain an artificial brain. MIT's Information Mechanics Group (Toffoli and Margolus) believe that within a few years it will be technically possible to update a trillion CA cells in about 0.1 nanoseconds (p221, Toffoli & Margolus 1990). Thus, if CA state transition rules can be found to make CA behave like neural networks, and if such CA based networks prove to be readily evolvable, then a potentially revolutionary new technology becomes possible. The CAM-Brain Project is based on the above ideas and fully intends to build artificial brains before the completion of the project in 2001. The potential is felt to be so great that it is likely that a new specialty will be formed, called "Brain Building".

For the first 18 months of the CAM-Brain Project, the author simulated a two dimensional version of CAM-Brain on a Sparc 10 workstation. This work was completed in the summer of 1994. The 2D version was used briefly (before work on the 3D version was started) to undertake some evolutionary tests, whose results will be presented in the next section. The 2D version served only as a feasibility and educational device. Since trails are obliged to collide in 2D, the 2D version was not taken very seriously. Work was begun rather quickly on the more interesting 3D version almost immediately after the 2D version was ready. Proper evolutionary tests will be undertaken once the 3D version is ready, which should be by early 1996. To begin to understand how cellular automata (Codd 1968) can be used as the basis for the growth and evolution of neural networks, consider Fig. 1 which shows an example of a 2D CA state transition rule, and Fig. 2 which shows a 2D CA trail, 3 cells wide. All cells in a CA system update the state of their cells

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

synchronously. The new state of a given cell depends upon its present state and the states of its nearest neighbors. Down the middle of the 3 cell wide CA trail, move "signal or growth cells" as shown in Fig. 2 As an example of a state transition rule which makes a signal cell move to the right one square, consider the right hand most signal cell in Fig. 2, which has a state of 5. The cell immediately to its right has a state of 1, which we want to become a 5. Therefore the 2D state transition rule to turn the 1 into a 5 is 1.2.2.2.5-->5. These signal or growth cells are used to generate the CA trails, by causing them to extend, turn left or right, split left or right, and Tsplit. When trails collide, they can form synapses. It is the sequence of these signal cells which determines the configuration of the CA trails, thus forming a CA network. It is these CA trails which later are used as neural network trails of axons and dendrites. Neural signals are sent down the middle of these CA trails. Thus there are two major phases in this process. Firstly, the CA trails are grown, using the sequence of signal cells. Secondly, the resulting CA trail network is used as a neural network, whose fitness at controlling some system can be measured and used to evolve the original growth sequence. To make this more explicit, it is the sequence of growth cells which is evolved. By modifying the sequence, one alters the CA network configuration, and hence the fitness of the configuration when it functions as a neural net in the second phase. From a genetic algorithm (GA) point of view, the format of the GA "chromosome" is the sequence of integers which code for the signaling or growth instructions. By mutating and crossing over these integers, one obtains new CA networks, and hence new neural networks. By performing this growth at electronic speeds in CAMs, and in parallel, with one CAM per GA chromosome, and attaching a conventional programmable microprocessor to each CAM to measure the user defined fitness of the CA based neural circuit, one has a means to evolve large numbers of neural modules very quickly. Using CAMs to evolve neural circuits, is an example of a type of machine that the author labels a "Darwin Machine", i.e. one which evolves its own structure or architecture. A related idea of the author concerns the concept of "Evolvable Hardware (EHW)" (de Garis 1993) where the software instructions used to configure programmable logic devices (PLDs) are treated as chromosomes in a Genetic Algorithm (Goldberg 1989). One then rewrites the circuit for each chromosome.

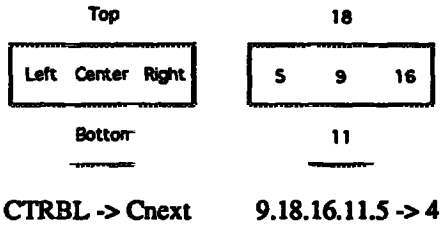

Fig. 1 A 2D CA State Transition Rule

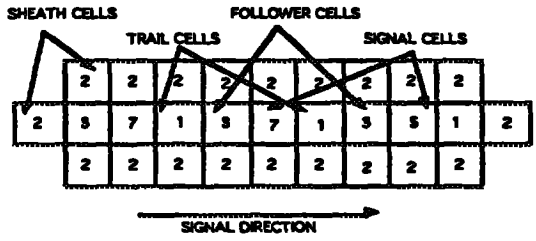

Fig. 2 Signal Cells Move Along a Cellular Automata Trail

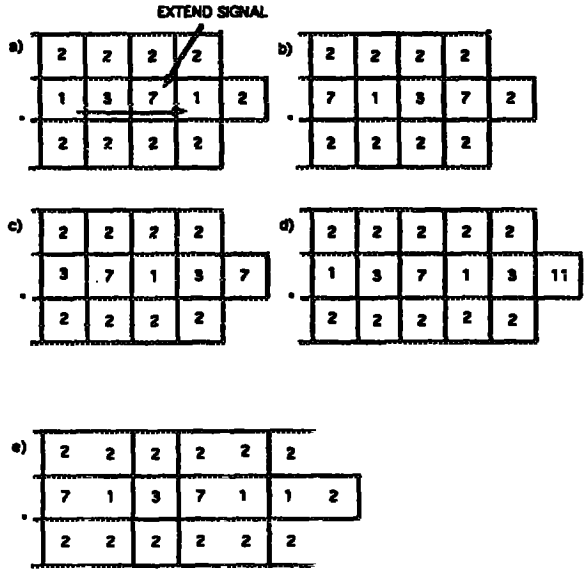

Fig. 3 Extend the Trail

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

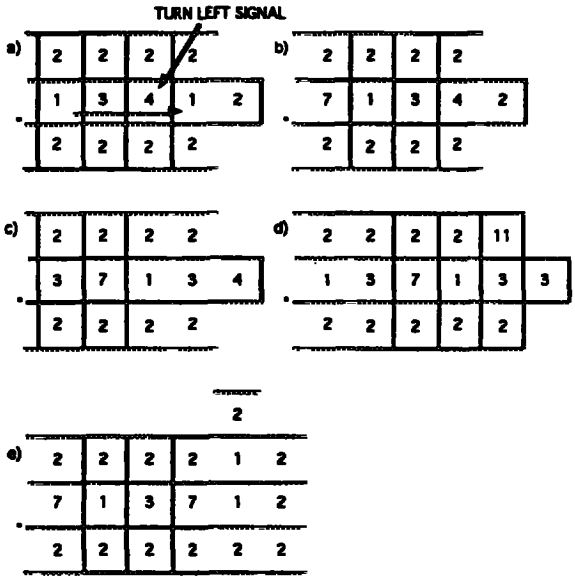

Fig. 4 Turn Trail Left

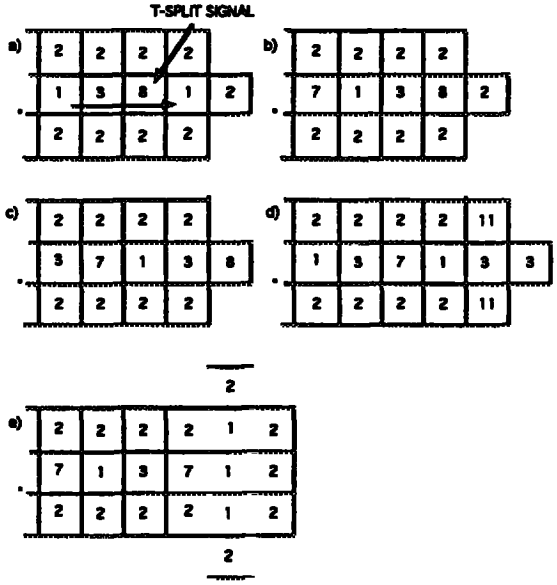

Fig. 6 T-Split Trail

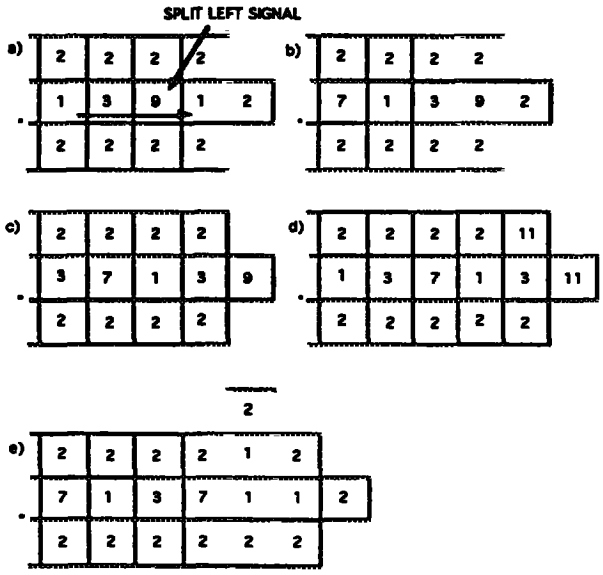

Fig. 5 Split Trail Left

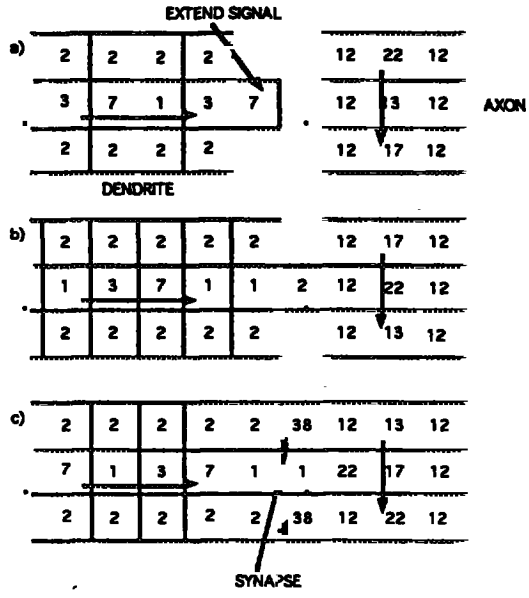

Fig. 7 Dendrite to Axon Synapsing

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

4. Further Details

This section provides further details on the implementation of the CA based neural networks. There are three kinds of CA trails in CAM-Brain, labeled dendrites, excitatory axons and inhibitory axons, each with their own states. Whenever an axon collides with a dendrite or vice versa, a "synapse" is formed. When a dendrite hits an excitatory/inhibitory axon or vice versa, an excitatory/inhibitory synapse is formed. An inhibitory synapse reverses the sign of the neural signal value passing through it. An excitatory synapse leaves the sign unchanged. Neural signal values range between -240 and +240 (or their equivalent CA states, ranging from 100 to 580). The value of a neural signal remains unchanged as it moves along an axon, but as soon as it crosses a synapse into a dendrite, the signal value (i.e. signal strength) begins to drop off linearly with the distance it has to travel to its receiving neuron. Hence the signal strength is proportional to the distance between the synapse and the receiving neuron. Thus the reduction in signal strength acts like a weighting of the signal by the time it reaches the neuron. But, this distance is evolvable, hence indirectly, the weighting is evolvable. CAM-Brain is therefore equivalent to a conventional artificial neural network, with its weighted sums of neural signal strengths. However, in CAM-Brain there are time delays, as signals flow through the network. When two or three dendrite signals collide, they sum their signal strengths (within saturated upper or lower bounds).

When implementing the 2D version of CAM-Brain, it soon became noticeable that there were many many ways in which collisions between CA trails could occur. So many, that the author became increasingly discouraged. It looked as though it would take years of handcoding the CA state transition rules to get CAM-Brain to work. The intention was to have rules which would cover every collision possibility. Eventually a decision was made to impose constraints on the ways in which CA trails could grow. The first such constraint was to make the trails grow on a grid 6 cells or squares (cubes) on a side. This process (called "gridding") sharply reduces the number of collision types. It also has a number of positive side effects. One is that in the neural signaling phase, neural signals arrive synchronously at junction points. One no longer needs to have to handcode rules for phase delays in neural signaling summation. By further imposing that different growth cells advance the length of the trails by the same number of squares, one can further reduce the number of collision types. With synchrony of growth and synchrony of signaling and gridding, it is possible to cover all possible types of collisions. Nevertheless, it still took over 11000 rules to achieve this goal, and this was only for the 2D version. The 3D version is expected to take about 150,000 rules, but due to the experience gained in working on the 2D version, and to the creation of certain software productivity tools, the 3D version should be completed by early 1996

Considering the fact that the 2D version takes 11,000 rules, it is impossible in this short work to discuss all the many tricks and strategies that are used to get CAM-Brain to work. That would require a book (something the author is thinking seriously about writing, if he ever makes time to do it). However, some of the tricks will be mentioned here. One is the frequent use of "gating cells", i.e. cells which indicate the direction that dendrite signals should turn at junctions to head towards the receiving neuron. To give these gating cells a directionality, e.g. a "leftness" or a "rightness", special marker cells are circulated at the last minute, after the circuit growth is stabilized. Since some trails are longer than others, a sequence of delay cells are sent through the network after the growth cells and before the marker cells. Without the delay cells, it is possible that the marker cells pass before synapses are formed.

Once the 2D simulation was completed (before the CAM8 was delivered) several brief evolutionary experiments using the 2D version were undertaken. The first, was to see if it would be possible to evolve the number of synapses. Figs. 9, 10, 11 show the results of an elite chromosome evolved to give a large number of synapses. Fig. 9 shows early growth. Fig. 10 shows completed growth, and Fig. 11 shows the neural signaling phase. In this experiment, the number of synapses increased steadily. It evolved successfully. The next experiment was to use the neural signaling to see if an output signal (tapped from the output of one of the neurons) could evolve to give a desired constant value. This evolved perfectly. Next, was to evolve an oscillator of a given arbitrary frequency and amplitude, which did evolve, but slowly (it took a full day on a Sparc10 workstation). Finally, a simple retina was evolved which output the two component directional velocity of a moving "line" which passed (in various directions) over a grid of 16 "retinal neurons". This also evolved but even more slowly. The need for greater speed is obvious.

The above experiments are only the beginning. The author has already evolved (not using CAs) the weights of recurrent neural networks as controllers of an artificial nervous system for a simulated quadrupedal artificial creature. Neural modules called "GenNets" (de Garis 1990, 1991) were evolved to make the creature walk straight, turn left or right, peck at food, and mate. GenNets were also evolved to detect signal frequencies, to generate signal frequencies, to detect signal strengths, and signal strength differences. By using the output of the detector GenNets, it was possible to switch motion behaviors. Each behavior had its own separately evolved GenNet. By switching between a library of GenNets (i.e. their corresponding evolved weights) it was possible to get the artificial creature to behave in interesting ways.

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

It could detect the presence and location of prey, predators and mates and take appropriate action, e.g. orientate, approach, and eat or mate, or turn away and flee. However, every time the author added another GenNet, the motion of the simulated creature slowed on the screen. The author's dream of being able to give a robot kitten some thousand different behaviors using GenNets, could not be realized on a standard monoprocessor workstation. Something more radical would be needed. Hence the motivation behind the CAM-Brain Project.

5. A Billion Neurons in a Trillion Cell CAM by 2001

Fig. 8a shows some estimated evolution times for 10 chromosomes over 100 generations for a Sparc 10 workstation, a CAM8, and a CAM2001 (i.e. a CAM using the anticipated electronics of the year 2001) for a given application. In the 2D version of CAM-Brain, implemented on a Sun Sparc 10 workstation, it takes approximately 3.4 minutes to grow a stable cellular automata network consisting of only four neurons. It takes an additional 3.2 minutes to perform the signaling on the grown network, i.e. a total growth-signaling time to measure the fitness of a chromosome of 6.6 minutes. This time scales linearly with the number of artificial neurons in the network. If one uses a population of 10 chromosomes, for 100 generations, the total evolution time (on a Sparc 10) is 100*10*6.6 minutes, i.e., 110 hours, or 4.6 days. This is obviously tediously slow, hence the need to use a CAM. MIT's CAM8 (Toffoli & Margolus 1990) can update 25 million cellular automata cells per second, per hardware module. A CAM8 "box" (of personal computer size) contains eight such modules, and costs about $40,000. Such boxes can be connected blockwise indefinitely, with a linear increase in processing capacity. Assuming an eight module box, how quickly can the above evolution (i.e. 100 generations, with a population size of 10) be performed? With eight modules, 200 million cell updates per second is possible. If one assumes that the 2D CA space in which the evolution takes place is a square of 100 cells on a side, i.e., 10,000 cells, then all of these cells can be (sequentially) updated by the CAM8 box in 50 microseconds. Assuming 1000 CA clock cycles for the growth and signaling, it will take 50 milliseconds to grow and measure the fitness of one chromosome. With a population of 10, and 100 generations, total CAM8 evolution time for a four neuron network will be 50 seconds, i.e. about one minute, which is roughly 8000 times faster. Using the same CAM8 box, and a 3D space of a million cells, i.e. a cube of 100 cells on a side, one could place roughly 40 neurons. The evolution time will be 100 times as long with a single CAM8 box. With 10 boxes, each with a separate microprocessor attached, to measure the fitness of the evolved network, the evolution time would be about eight minutes. Thus for 1000 neurons, the evolution would take about 3.5 hours, quite an acceptable figure. For a million neurons, the evolution time would be nearly five months. This is still a workable figure. Note, of course, that these estimates are lower bounds. They do not include the necessary human thinking time, and the time needed for sequential, incremental evolution, etc. However, since the CAM-Brain research project will continue until the year 2001, we can anticipate an improvement in the speed and density of electronics over that period. Assuming a continuation of the historical doubling of electronic component density and speed every two years, then over the next eight years, there will be a 16-fold increase in speed and density. Thus the "CAM-2001" box will be able to update at a rate of 200*16*16 million cells per second. To evolve the million neurons above will take roughly 13.6 hours. Thus to evolve a billion neurons, will take about 19 months, again a workable figure. But, if a million neurons can be successfully evolved, it is likely that considerable interest will be focused upon the CAM-Brain approach, so that more and better machines will be devoted to the task, thus reducing the above 19-month figure. For example, with 100 machines, the figure would be about two months. The above estimates are summarized in Figure 8a. These estimates raise some tantalizing questions. For example, if it is possible to evolve the connections between a billion artificial neurons in a CAM2001, then what would one want to do with such an artificial nervous system (or artificial brain)? Even evolving a thousand neurons raises the same question.

Sparc10	CAM8	CAM8	CAM8	CAM8	CAM2001	CAM2001
10000 CA cells	10000 CA cells	1 million CA cells	25 million CA cells	25 billion CA cells	25 billion CA cells	25 trillion CA cells
4 neurons	4 neurons	40 neurons	1000 neurons	1 million neurons	1 million neurons	1 billion neurons
1 Sparc10	1 CAM8	10 CAM8s	10 CAM8s	10 CAM8s	10 CAM2001s	100 CAM2001s
4.6 days	50 seconds	8 minutes	3.5 hours	5 months	13.6 hours	2 months

Fig. 8a Evolution Times for Different Machines & CA Cell, Neuron & Machine Numbers

48*48*24	10gens 51	40gens 63	60gens 71	100gens 93	
96*48*24	10gens 81	20gens 89	45gens 122		
96*96*24	5gens 116	10gens 116	40gens 205	45gens 205	70gens 234
96*96*48	5gens 235	10gens 235			

Fig. 8b Synapses per Neuron Doubles as 3D Space Doubles

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

One of the aims of the CAM-Brain research project is to build an artificial brain which can control 1000 behaviors of a "robot kitten" (i.e. a robot of size and capacities comparable to a kitten) or to control a household "cleaner robot". Presumably it will not be practical to evolve all these behaviors at once. Most likely they will have to be evolved incrementally, i.e., starting off with a very basic behavioral repertoire and then adding (stepwise) new behaviors. In brain circuitry terms, this means that the new neural modules will have to connect up to already established neural circuits. In practice, one can imagine placing neural bodies (somas) external to the established nervous system and then evolving new axonal and dendral connections to it.

The CAM-Brain Project hopes to create a new tool to enable serious investigation of the new field of "incremental evolution." This field is still rather virgin territory at the time of writing. This incremental evolution could benefit from using embryological ideas. For example, single seeder cells can be positioned in the 3D CA space under evolutionary control. Using handcrafted CA "developmental or embryological" rules, these seeder cells can grow into neurons ready to emit dendrites and axons (de Garis 1992). The CAM-Brain Project, if successful, should also have a major impact on both the field of neural networks and the electronics industry. The traditional preoccupation of most research papers on neural networks is on analysis, but the complexities of CAM-Brain neural circuits, will make such analysis impractical. However, using Evolutionary Engineering, one can at least build/evolve functional systems. The electronics industry will be given a new paradigm, i.e. evolving/growing circuits, rather than designing them. The long term impact of this idea should be significant, both conceptually and financially.

6. The 3D Version

The 3D version is a conceptually (but not practically) simple extension of the 2D version. Instead of 4 neighbors, there are 6 (i.e. North, East, West, South, Top, Bottom). Instead of 6 growth instructions as in the 2D version (i.e. extend, turn left, turn right, split extend left, split extend right, split left right), there are 15 in the 3D version. A 3D CA trail cross section consists of a center cell and 4 neighbor cells, each of different state or color (e.g. red, green, blue, brown). Instead of a turn left instruction being used as in the 2D case, a "turn green" instruction is used in the 3D case. The 15 3D growth instructions are (extend, turn red, turn green, turn blue, turn brown, split extend red, split extend green, split extend blue, split extend brown, split red brown, split red blue, split red green, split brown blue, split brown green, split blue green). A 3D CA rule thus consists of 8 integers of the form CTSENWB-->Cnew. The 3D version enables dendrites and axons to grow past each other, and hence reach greater distances. The weakness with the 2D version is that collisions in a plane are inevitable, which causes a crowding effect, whereby an axon or dendrite cannot escape from its local environment. This is not the case with the 3D version, which is topologically quite different. A 3D version is essential if one wants to build artificial brains with many interconnected neural modules. The interconnectivity requires long axons/dendrites. Fig. 12 shows an early result in 3D simulation. A space of 3D CA cells (48*48*48 cubes) was used. A single short 3D CA trail was allowed to grow to saturate the space. One can already sense the potential complexity of the neural circuits that CAM-Brain will be able to build. In 3D, it is likely that each neuron will have hundreds, maybe thousands of synapses, thus making the circuits highly evolvable due to their smooth fitness landscapes (i.e. if you cut one synapse, the effect is minimal when there are hundreds of them per neuron).

7. Recent Work

Just prior to writing this work, the author was able to test the idea that in 3D a single neuron could have an arbitrarily large number of synapses, provided that there is enough space for them to grow in. This was a crucial test, whose results are shown in Fig. 8b. Fitness was defined as the number of synapses formed for two neurons in CA spaces of 48*48*24, 96*48*24, 96*96*24, and 96*96*48 cells respectively. One can see that by doubling the space, one doubles (roughly) the number of synapses (for the elite chromosome). If this had not been the case, for example, if some kind of fractal effect had caused a crowding of the 3D circuits (similar to the crowding effect in 2D), then the whole CAM-Brain project would have been made doubtful. However, with this result, it looks as though evolvability in the 3D signaling phase will be excellent, although the author needs several months more work before completing the 3D signaling phase to confirm his confidence.

At the time of writing (December 1995), the author is completing the simulation of the 3D version, working on the many thousands of rules necessary to specify the creation of synapses. So far, more than 140,000 3D rules have been implemented, and it is quite probable that the figure may go higher than 150,000. Since each rule is rotated 24 ways (6 ways to place a cube on a surface, then 4 ways to rotate that cube) to cater to all possible orientations of a 3D trail, the actual number of rules placed in the (hashed) rule base will be over 3 million. Specifying these rules takes time, and constitutes so far, the bulk of the effort spent building the CAM-Brain system. Software has been written to help automate this rule generation process, but it remains a very time consuming business. Hence the immediate future work will be to complete the simulation of the 3D version. Probably, this will be done by early 1996.

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

Early in 1995, the author put his first application on the CAM8 machine (which rests on his desk). MIT's CAM8 is basically a hardware version of a look up table, where the output is a 16 bit word which becomes an address in the look up table at the next clock cycle. This one clock cycle lookup loop is the reason for CAM8's speed. It is possible to give each CA cell in the CAM8 more than 16 bits, but tricks are necessary. The first CAM8 experiment the author undertook involved only 16 bits per CA cell. This work is too short to go into details as to how the CAM8 functions, so only a broad overview will be given here. The 16 bits can be divided into slices, one slice per neighbor cell. These slices can then be "shifted" (by adding a displacement pointer) by arbitrarily large amounts (thus CAM8 CA cells are not restricted to having local neighbors). With only 16 bits, and 4 neighbors in the 2D case (Top, Right, Bottom, Left) and the Center cell, that's only 3 bits per cell (i.e. 8 states, i.e. 8 colors on the display screen). It is not possible to implement CAM-Brain with only 3 bits per CA cell. It was the intention of the author to use the CAM8 to show its potential to evolve neural circuits with a huge number of artificial neurons. The author chose an initial state in the form of a square CA trail with 4 extended edges. As the signals loop around the square, they duplicate at the corners. Thus the infinite looping of 3 kinds of growth signals supply an infinite number of growth signals to a growing CA network. There are 3 growth signals (extend, extend and split left, extend and split right). The structure needs exactly 8 states. The 8 state network grows into the 32 megacells of 16 bits each, which are available in the CAM8. At one pixel per cell, this 2D space takes over 4 square meters of paper poster (hanging on the author's wall). A single artificial neuron can be put into the space of one's little finger nail, thus allowing 25,000 neurons to fit into the space. If 16 Mbit memory chips are used instead of 4 Mbit chips, then the area and the number of neurons quadruples to 100,000.

Placing the poster on the author's wall suddenly gave visitors a sense of what is to come. They could see that soon a methodology will be ready which will allow the growth and evolution of artificial brains, because soon it will be possible to evolve many thousands of neural modules and their inter-connections. The visitors sense the excitement of CAM-Brain's potential.

Filling a space of 32 Mcells, with artificial neurons can be undertaken in at least two ways. One is to use a very large initialization table with position vectors and states. Another, is to allow the neurons to "grow" within the space. The author chose to use this "neuro-embryonic" approach. A single "seeder" CA cell is placed in the space. This seeder cell launches a cell to its right and beneath it. These two launched cells then move in their respective directions, cycling through a few dozen states When the cycle is complete, they deposit a cell which grows into the original artificial neuron shape that the author uses in the 2D version of CAM-Brain. Meanwhile other cells are launched to continue the growth. Thus the 32Mcell space can be filled with artificial neurons ready to accept growth cell "chromosomes" to grow the neural circuitry. This neuro-embryogenetic program (called "CAM-Bryo") was implemented on a workstation by the author, and ported to the CAM8 by his research colleague Felix Gers. In order to achieve the porting, use was made of "subcells" in the CAM8, a trick which allows more than 16 bits per CA cell, but for N subcells of 16 bits, the total CAM8 memory space available for CA states is reduced by a factor of N. Gers used two subcells for CAM-Bryo, hence 16M cells of 32 bits each. A second poster of roughly two square meters was made, which contained about 25,000 artificial neurons (see Fig. 13). Again, with 16Mbit memory chips, this figure would be 100,000. Gers expects to be able to port the 2D version of CAM-Brain to the CAM8 with a few weeks work, in which case, a third poster will be made which will depict about 15,000 neurons (with a lower density, to provide enough space for the neural circuitry to grow) and a mass of complex neural circuits. Once this is accomplished, we expect that the world will sit up and take notice - more on this in the next section.

The author's boss at ATR's Evolutionary Systems department, has recently set up a similar group at his company NTT, called Evolutionary Technologies (ET) department. The idea is that once the ATR Brain Builder group's research principles are fairly solid, the author and the author's boss (whose careers are now closely linked) will be able to tap into the great research and development resources of one of the world's biggest companies, when the time comes to build large scale artificial brains. NTT has literally thousands of researchers.

The author would like to see Japan invest in a major national research project within the next 10 years to build "Japan's Artificial Brain", the so-called "J-Brain Project". This is the goal of the author, and then to see such a project develop into a major industry within 20 years. Every household would like to have a cleaner robot controlled by an artificial brain. The potential market is huge.

8. Future Work

A lot of work remains to be done. The author has a list of "to dos" attached to his computer screen. The first item on the list is of course, to finish the rules for the 3D version of CAM-Brain. This should be done by early 1996, and will probably need over 150,000 CA rules. Second, the experience gained in porting the 700 rules for "CAM-Bryo" from a workstation to the CAM8 will shortly enable Gers to complete the much tougher task of porting the 2D version of CAM-Brain to the CAM8. In theory, since

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

there are 11,000 CA rules for the 2D version, and that each rule has 4 symmetry rotations, that makes about 45,000 rules in total to be ported. This fits into the 64K words addressable by 16 bits. The 3D version however, with its (estimated) 150,000 rules, and its 24 symmetry rotations, will require over 3 million rules in total. The 3D version may require a "Super CAM" to be designed and built (by NTT's "Evolutionary Technologies" Dept., with whom the author collaborates closely), which can handle a much larger number of bits than 16. The group at MIT who built CAM8 is thinking of building a CAM9 with 32 bits. This would be very interesting to the author. Whether NTT or MIT get there first, such a machine may be needed to put the 3D version into a CAM. However, with a state-of-the-art workstation (e.g. a DEC Alpha, which the user has on his desk) and a lot of memory (e.g. 256 Mbyte RAM), it will still be possible to perform some interesting evolutionary experiments in 3D CAM-Brain, but not with the speed of a CAM.

Another possibility for porting the 3D version to the CAM8, is to re implement it using CA rules which are more similar to those used in von Neumann's universal constructor/calculator, rather than Codd's. Von Neummann's 2D trails are only 1 cell wide, whereas Codd's 2D version are 3 cells wide, with the central message trail being surrounded by two sheath cells. The trick to using von Neumann's approach is incorporating the direction of motion of the cell as part of the state. The author's colleague Jacqueline Signorinni advises that CAM-Brain could be implemented at a higher density (i.e. more filled CA cells in the CA space) and without the use of a lookup table. The control of the new states would be implemented far more simply she feels, by simple IF-THEN-ELSE type programming. "von Neumann-izing" the 3D version of CAM-Brain might be a good task for the author's next grad student.

With the benefit of hindsight, if the 3D version is reimplemeted (and it is quite likely that my boss will have other members of our group do just that), then the author would advise the following. If possible (if you are implementing a Codd version) give the four sheath cells in a 3D CA trail cross section the same state. This would obviously simplify the combinatorial explosion of the number of collision cases during synapse formation. But, how then would the 3D growth instructions be interpreted when they hit the end of a trail, and how would you define the symmetry rotations? If possible, it would also be advisable to use the minimum number of gating cell states at growth junctions for all growth instructions. Whether this is possible or not, remains to be seen. However, if these simplifications can be implemented (and of course the author thought of them originally, but was unable to find solutions easily), then it is possible that the number of 3D CA rules might be small enough to be portable to the CAM8, which would allow 3D neural circuits to be evolved at 200 million CA cells per second (actually less because of the subcell phenomenon).

Once the 3D rules are ready, two immediate things need to be done. One is to ask ATR's graphics people to display these 3D neural circuits in an interesting, colorful way, perhaps with VR (virtual reality) 3D goggles with interactivity and zoom, so that viewers can explore regions of the dynamic circuits in all their 500 colors (states). This could be both fun and impressive. The second thing is of course to perform some experiments on the 3D version. As mentioned earlier, this will have to be done on a workstation, until a SuperCAM is built. Another possibility, as mentioned earlier is to redesign the 3D CA rules, to simplify them and reduce their number so that they can fit within the 64K 16 bit confines of the CAM8 machine.

As soon as the 2D rules have been fully ported to the CAM8, experiments can begin at speed. Admittedly the 2D version is topologically different from the 3D version (in the sense that collisions in 2D are easier than in 3D), it will be interesting to try to build up a rather large neural system with a large number of evolved modules (e.g. of the order of a hundred, to start with). At this stage, a host of new questions arise. Look at Fig. 14, which is van Essen's famous diagram of the modular architecture of the monkey's visual and motor cortex, showing how the various geographical regions of the brain (which correspond to the rectangles in the figure, and to distinct signal processing functions) connect with each other. Physiological techniques now exist which enable neuro-anatomists to know which distinct cortical regions connect to others. Thus the geography (or statics) of the biological brain is increasingly known. What remains mysterious of course, is the dynamics. How does the brain function.

Van Essen's diagram is inspirational to the author. The author would like to produce something similar with CAM-Brain, i.e. by evolving neural modules (corresponding to the rectangles, or parts of the rectangles) and their interconnections. This raises other questions about sequencing and control. For example, does one evolve one module and freeze its circuits and then evolve another module, freeze its circuits and finally evolve the connections between them, or does one evolve the two modules together, or what? Will it be necessary to place walls around each module, except for hand crafted I/O trails? The author has no clear answers or experience yet in these matters. The author's philosophy is "first build the tool, and then play with it. The answers will come with using the tool".

Another possibility for future work is to try to simplify the whole process of rule making. Perhaps higher level rules can be made which are far fewer in number and allow the author's low level rules to be generated from

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

them. If such a thing can be done, it would be nice, but the author believes there are still so many special cases in the specification of 3D CAM-Brain, that the number of high level rules may still be substantial. If these high level rules can be found, it might be possible to use them and put them on the CAM8, so that 3D evolutionary experiments can be undertaken at CAM8 speeds. Another idea is to use FPGAs (field programmable gate arrays) which code these high level rules and then to use them to grow 3D neural circuits. Each 3D CA cell could contain pointers to its 3D neighbors. In this way, it would be possible to map 3D neural circuits onto 2D FPGAs. This is longer term work. FPGAs are not cheap if many are needed. The author's RAM based solution has the advantage of being cheap, allowing a billion (one byte) CA cell states to be stored reasonably cheaply.

A recent suggestion coming from NTT concerns the use of an existing "content addressable memory" machine, which may be able to update CA cells effectively. There is a "CAMemory" research group at NTT that ATR is now collaborating with. If a small enough number of CAMemory Boolean function rules corresponding to CAM-Brain can be found (a big if), it is possible that a NTT's CAMemory could be thousands of times faster than the CAM8. Obviously, such a possibility is worth investigating, and if successful, could be extremely exciting, since it would mean *hundreds of billions* of CA cell updates a second.

The author feels that the nature of his research in 1996 will change from one of doing mostly software simulation (i.e. generating masses of CA rules), to learning about the biological brain (i.e. reading about brain science to get ideas to put into CAM-Brain), hardware design, and evolvable hardware. These activities will proceed in parallel. Of course, evolutionary experiments, on CAM8 for the 2D version of CAM-Brain, and on a 256 Mbyte RAM (DEC Alpha) workstation for the 3D version, will also be undertaken in parallel.

Further down the road, will be the attempt to design a "nanoCAM" or "CAM2001" based on nanoelectronics. The Brain Builder Group at ATR is collaborating with an NTT researcher who wants to build nano-scale cellular automata machines. With the experience of designing and building a "SuperCAM", a nanoscale CAM should be buildable with several orders of magnitude greater performance. Further research aims are to use CAs to make Hebbian synapses capable of learning. One can also imagine the generation of artificial "embryos" inside a CA machine, by having CA rules which allow an embryological "unfolding" of cell groups, with differentiation, transportation, death, etc. resulting in a form of neuro-morphogenesis similar to the way in which biological brains are built. The author's "CAM-Bryo" program is an early example of this kind of neuro-morphogenetic research.

9. Summary

The CAM-Brain Project at ATR, Kyoto, Japan, intends to build/grow/evolve a cellular automata based artificial brain of a billion artificial neurons at (nano-)electronic speeds inside Cellular Automata Machines (CAMs) by the year 2001. Quoting from a paper by Margolus and Toffoli of MIT's Information Mechanics group, "We estimate that, with integrated circuit technology, a machine consisting of a trillion cells and having an update cycle of 100 pico-second for the entire space will be technologically feasible within 10 years" (i.e. by 2000) (Margolus and Toffoli 1990). In a trillion 3D CA cells (cubes), one can place billions of artificial neurons. Such an artificial nervous system will be too complex to be humanly designable, but it may be possible to evolve it, and incrementally, by adding neural modules to an already functional artificial nervous system. In the summer of 1994, a 2D simulation of CAM-Brain using over 11000 hand crafted CA state transition rules was completed, and initial tests showed the new system to be evolvable. By early 1996, a 3D simulation will be completed.

If the CAM-Brain Project is successful, it will revolutionize the field of neural networks and artificial life, because it will provide a powerful new tool to evolve artificial brains with billions of neurons, and at electronic speeds. The CAM-Brain Project will thus produce the first Darwin Machine, i.e. a machine which evolves its own architecture. The author is confident that in time a new specialty will be established, based partly on the ideas behind CAM-Brain. This specialty is called simply "Brain Building".

The author and his colleague Felix Gers are about to port the 2D version of CAM-Brain to the CAM8. Hence in early 1996, it will be possible to evolve neural circuits with 25,000 neurons (or 100,000 neurons, with 16 Mbit memory chips) at 200 million CA cell updates a second. As mentioned earlier, the author expects that when this happens, the world will sit up and take notice. Twenty years from now, the author envisages the brain builder industry (i.e. intelligent robots etc.) as being one of the world's top industries, comparable with oil, automobile, and construction. He sees an analogy between the efforts of the very early rocket pioneers (e.g. the American Goddard, and the German (V2) von Braun) and the US NASA mission to the moon which followed. Today's 100,000-neuron artificial brain is just the beginning of what is to come. With adiabatic (heat generationless) reversible quantum computation, it will be possible to build 3D hardware circuits that do not melt. Hence size becomes no obstacle, which means that one could use planetoid size asteroids to build huge 3D brain like computers containing ten to power 40 components with one bit per atom. Hence late into the 21st century, the author predicts that human beings will be confronted with

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

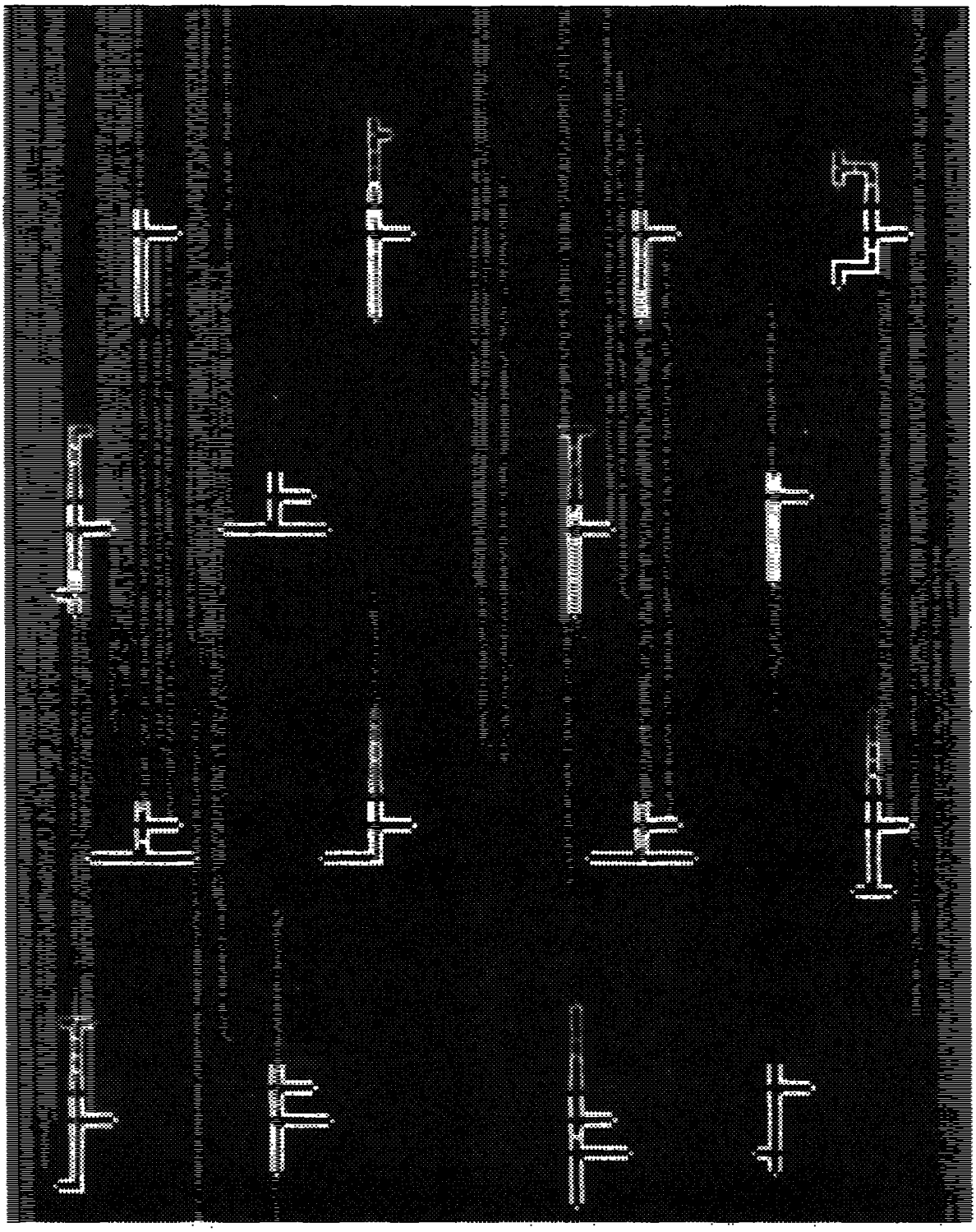

Fig. 9 2D CAM-Brain Early Growth

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

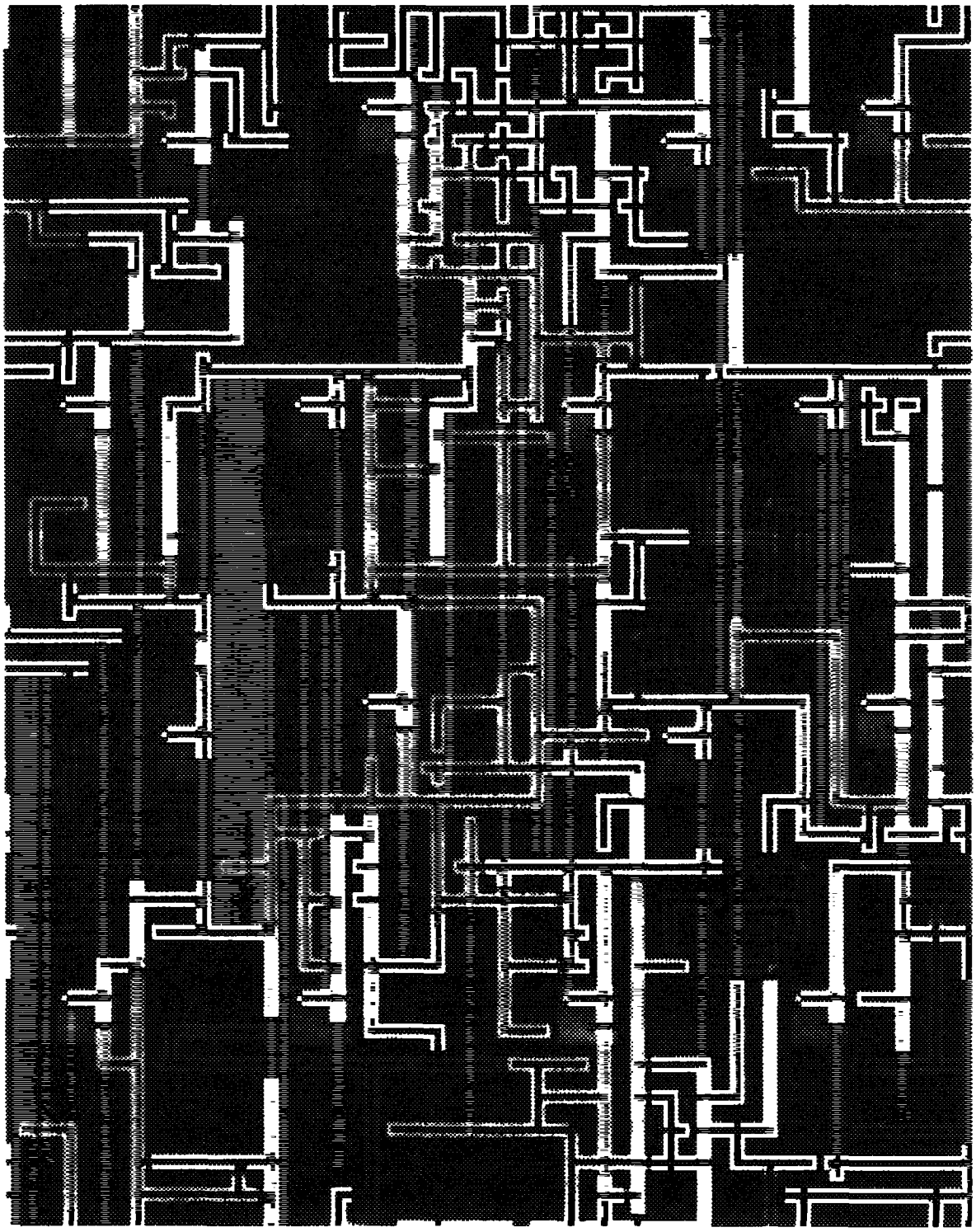

Fig. 10 2D CAM-Brain Completed Growth

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

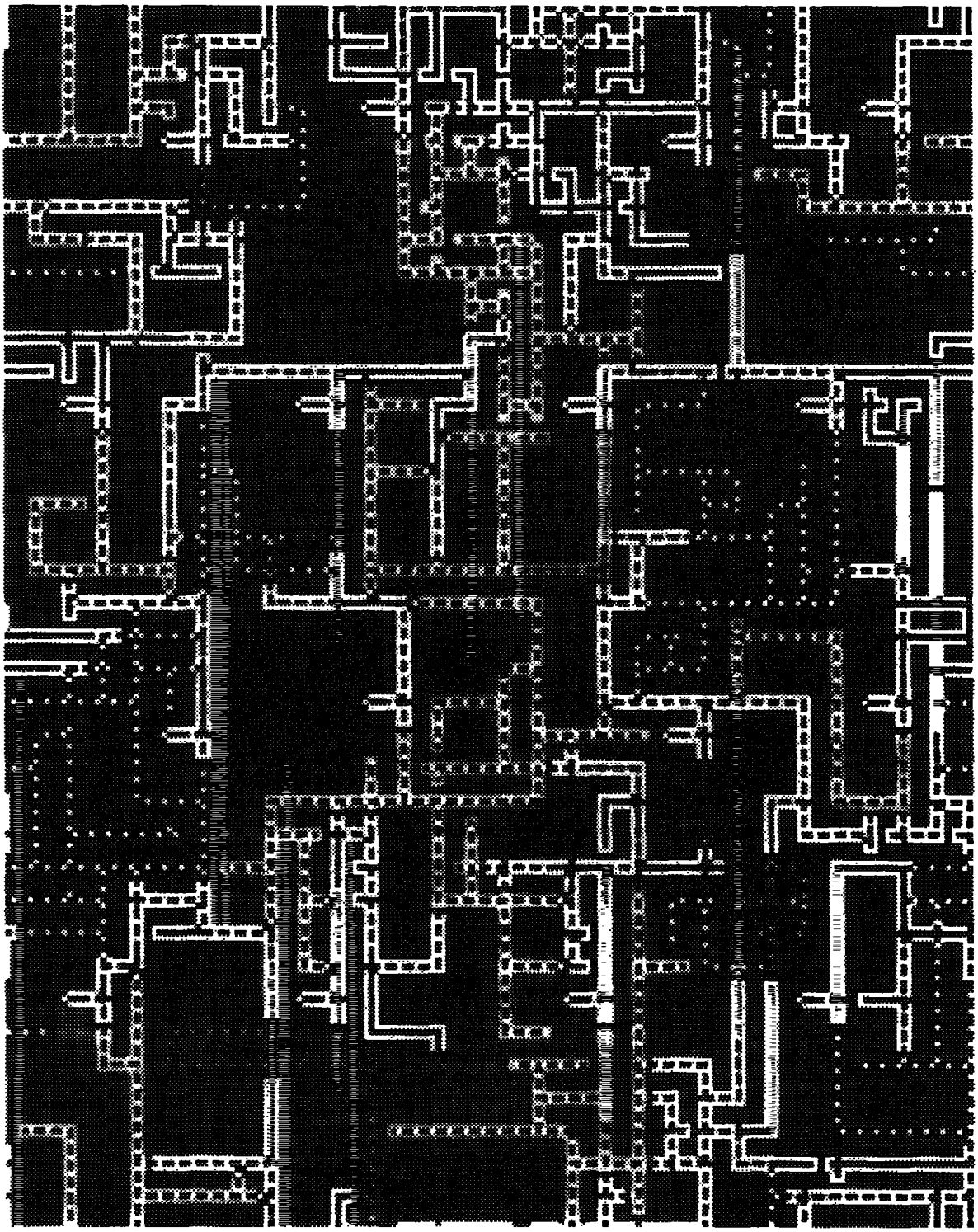

Fig. 11 2D CAM-Brain Neural Signaling

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

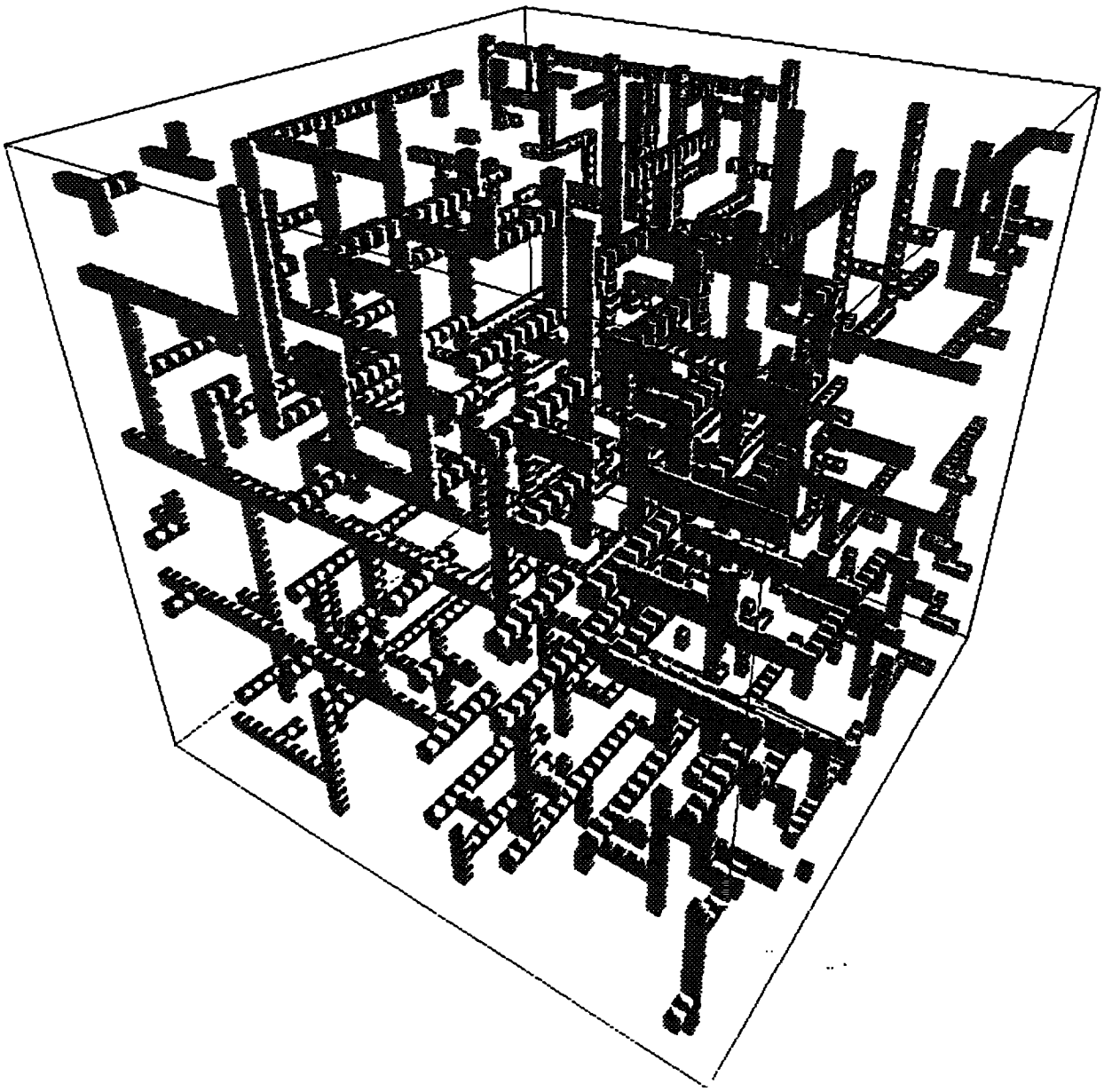

Fig. 12 3D CAM-Brain Non-Synaptic Growth

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

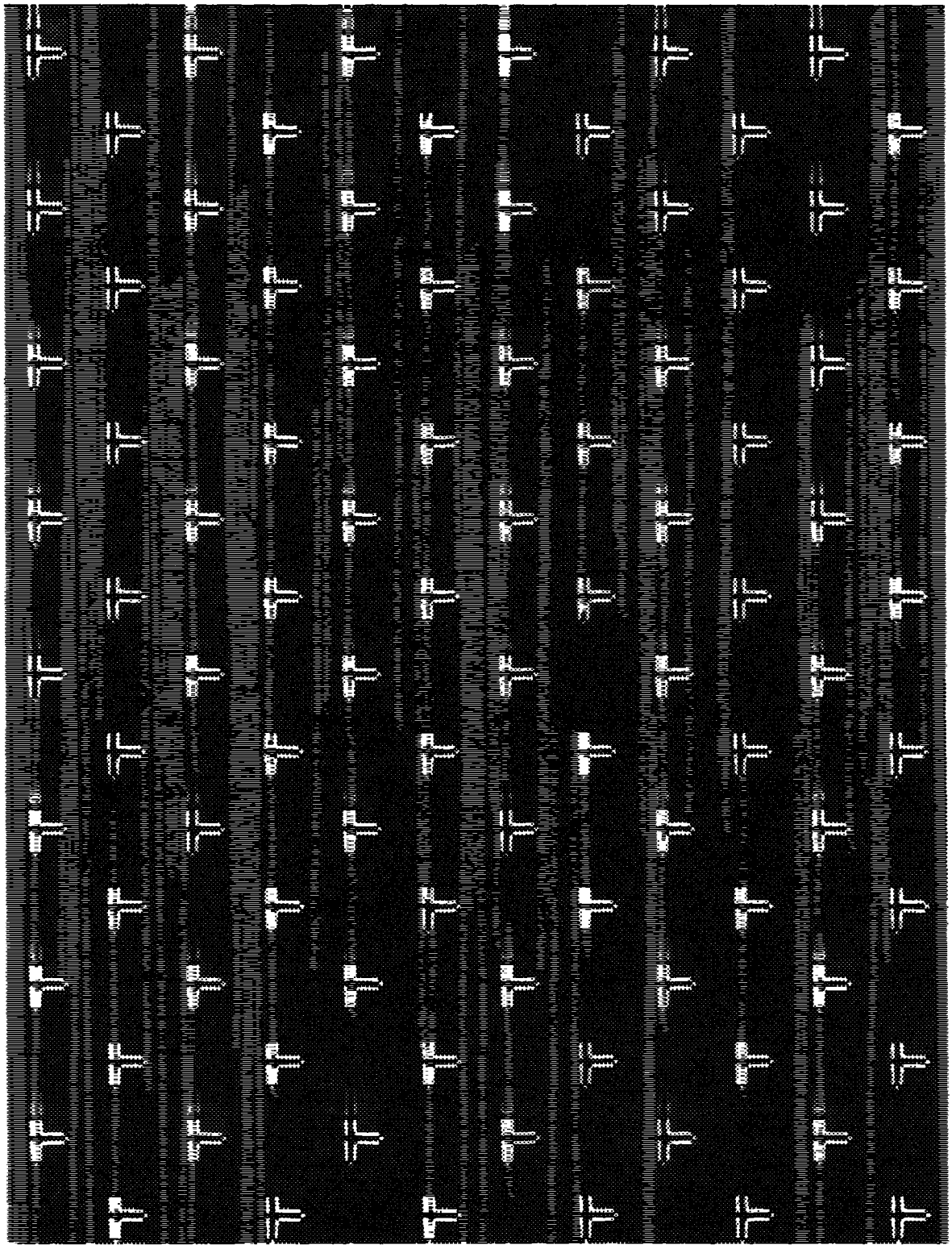

Fig. 13 2D CAM-Bryo

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

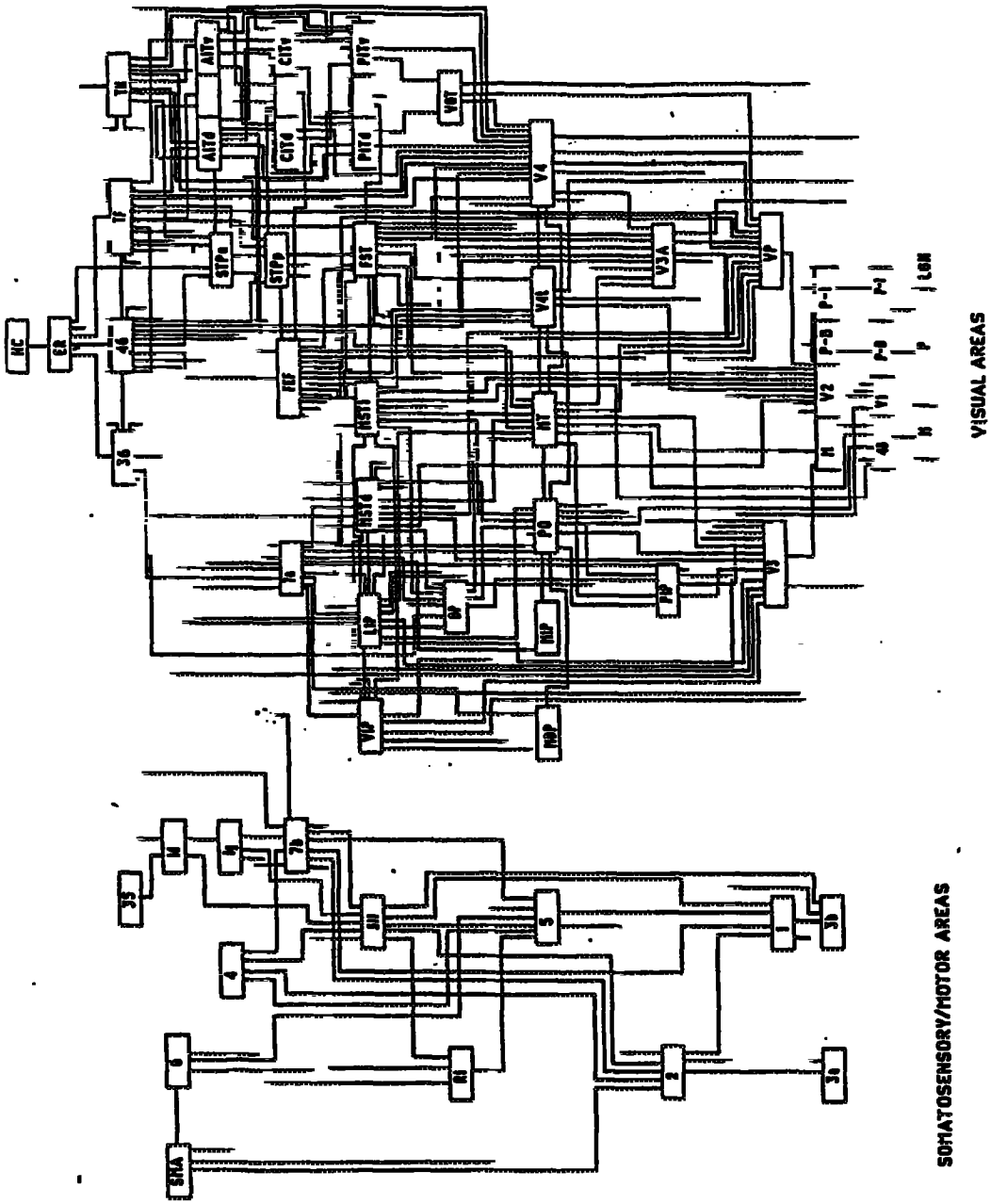

Fig. 14 van Essen's Monkey Brain Architecture

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

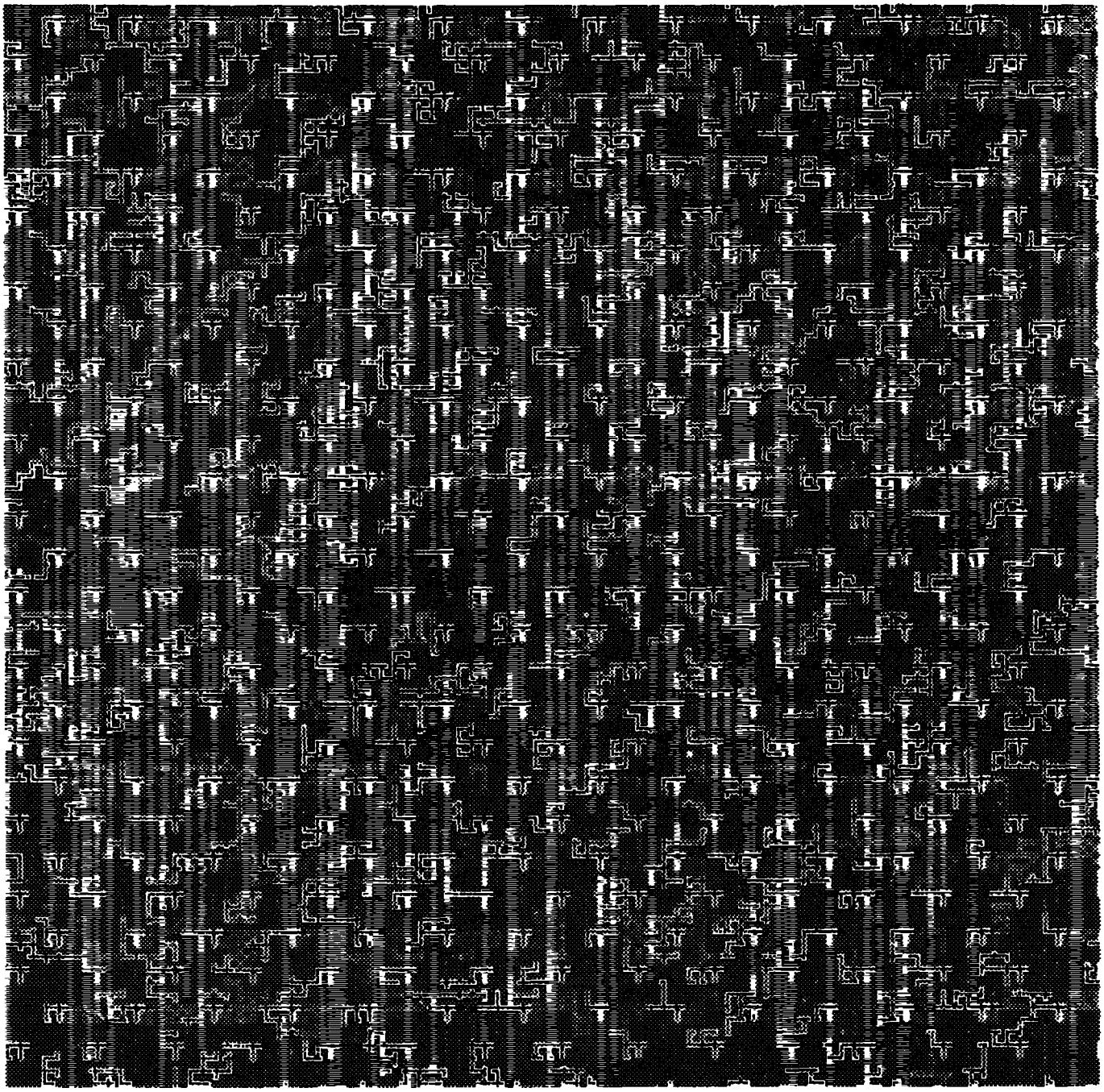

Fig. 15 2D CAM-Brain on MIT's "CAM-8"

From: Proceedings of the Third International Conference on Multistrategy Learning. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

the "artilect" (artificial intellect) with a brain vastly superior to the human brain with its pitiful trillion neurons. The issue of "species dominance" will dominate global politics late next century. The middle term prospects of brain building are exciting, but long term they are terrifying. The author has written an essay on this question (de Garis 1995). If you would like to be sent a copy, just email him at degaris@hip.atr.co.jp (The author will set up his home page on the web in 1996, after making the effort to learn html).

Finally, by way of a postscript - as the author was preparing the final draft, there were 6 people at ATR working on CAM-Brain (the author (3D CA rules), and his colleague Felix Gers (porting 2D to CAM-8), the author's Japanese colleague Hemmi and his programmer assistant Yoshikawa (translating CA rules to Boolean expressions), and two M. Sc. students from Nara Institute of Science and Technology (NAIST). At NTT, there were 3-4 people from the Content Addressable Memory machine group who were finding ways to apply their machine to CAM-Brain. So, things are certainly hotting up.

(Note added, April 1996) - Fig. 15 shows about 800 artificial neurons with their axons and dendrites grown using the CAM-8 machine with 128 Mega words of 16 bits. This figure is taken from an 8 square meter poster containing 100,000 neurons. In a year, this number will probably be a million. Felix Gers thinks he can port the 3D version to the CAM-8. The 3D rules are almost complete and number over 160,000, i.e. nearly 4 million with (24) rotations.

References

E.F. Codd, *Cellular Automata*, Academic Press, NY, 1968.

Hugo de Garis, "Genetic Programming: Modular Evolution for Darwin Machines,"*ICNN-90WASH-DC*, (Int. Joint Conf. on Neural Networks), January 1990, Washington DC, USA.

Hugo de Garis, "Genetic Programming", Ch.8 in book *Neural and Intelligent Systems Integration*, ed. Branko Soucek, Wiley, NY, 1991.

Hugo de Garis, "Artificial Embryology : The Genetic Programming of an Artificial Embryo", Ch.14 in book *Dynamic, Genetic, and Chaotic Programming*, ed. Branko Soucek and the IRIS Group, Wiley, NY, 1992.

Hugo de Garis, "Evolvable Hardware : Genetic Programming of a Darwin Machine", in *Artificial Neural Nets and Genetic Algorithms*, R.F. Albrecht, C.R. Reeves, N.C. Steele (eds.), Springer Verlag, NY, 1993.

Hugo de Garis, "Genetic Programming : Evolutionary Approaches to Multistrategy Learning", Ch.21 in book "Machine Learning : A Multistrategy Approach, Vol.4", R.S. Michalski & G. Tecuci (eds), Morgan Kauffman, 1994.

Hugo de Garis, "Cosmism : Nano Electronics and 21st Century Global Ideological Warfare", (to appear in a future nanotech book).

K.E. Drexler, *Nanosystems : Molecular Machinery, Manufacturing and Computation*, Wiley, NY, 1992.

D..E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

T. Toffoli & N. Margolus, *Cellular Automata Machines*, MIT Press, Cambridge, MA, 1987; and *Cellular Automata Machines*, in Lattice Gas Methods for Partial Differential Equations, SFISISOC, eds. Doolen et al, Addison-Wesley, 1990.