Computational Biology on Massively Parallel Machines

Klaus Schulten
Beckman-Institute and Department of Physics
University of Illinois
Urbana,IL 61801, USA

1 Introduction

Computational Methods have long been developed into useful tools in Science and Engineering. The complex nature of living systems has delayed such development in the life sciences; however, during the past decade Biomedical Research and Technology has seen an influx of computational methods equal to that in the physical sciences. An example is the widespread use of computer imaging techniques in medical diagnostics. Another example is the application of computational methods for drug design and structure refinement in Molecular Biology and Medicine. A third example is the emergence of the field of Computational Neural Science which attempts to understand the principles of development and functional cooperation of the neurons in the brain, using computer simulations as a main tool. There have been many reasons for the proliferation of computational techniques in Biology and Medicine, not the least of which has been the rapid development of the computer itself which became increasingly better adapted to the complex data processing tasks required in Biology and Medicine.

Hurdles in Computational Biology: Three Examples

The available computer power and the algorithms of Computational Biology are, in many cases, still inadequate. In Molecular Biology, for example, the numerical simulation of a protein of a few thousand atoms, over the time span of one μs, would require about 100 years even on a Cray 2. Simulations of biopolymers in natural surroundings, e.g., in a membrane and water, would involve 10^6 atoms and, presently, are still very difficult to achieve, even for very brief time spans.
The description of brain activity as observed, for example, through voltage-sensitive dyes in a cortical area of a few mm^2 involves 10^5 neurons with 10^8 synaptic connections. The simulation of the evolution of the connectivity scheme of brain areas with 10^8 dynamic variables, e.g., connections between the retinas of the eye and the striate cortex, requires the fastest modern computers, if possible at all.
Diagnostic techniques, e.g., Magnetic Resonance Imaging (MRI), are based on physical processes which need to be understood if the diagnostic method is to be developed further. The ultimate level of understanding is reached if the measuring process for a sample can be simulated in its entirety, a task which, in case of MRI, requires one to monitor the nuclear spin precession of 10^6 diffusing water molecules over many precession periods. Again, such simulation presently runs many days on the fastest computers.

Opportunities Through Massively Parallel Computers

The five years 1991-1995 will be the first period in the history of computing that experiences an increase in speed by three orders of magnitude in such a short time. This development, of course, is due to the emergence of massively parallel computers. To describe this development let us begin with a look at the state of the art in supercomputing by comparing the Gflop performance of this year's and next year's high end machines. For this purpose, we compare in Table 1 the performance of machines of four vendors running today (1991) and expected to be shipped next year (1992). Certainly, the performance for next year can be estimated only very crudely. (Performance figures for the new (1992) model of the Connection machine were left blank, but should be available by the time of the lecture.)

Table 1 shows several trends. First, the machines compared show a speed-up by a factor of 4 to 6 between 1991 and 1992. This speed up is achieved through use of a new processor, in case of the Transputer-based Parsytec machines (the T800 of the SC-400 is replaced by the T9000 of the GC-2), due to redesign of the machine in case of the Connection Machine CM-5, and due to increased processor numbers in case of Intel's Sigma Touchstone based on the i860 processor.

The manufacturers of the machines listed in Table 1 are expected to push the performance of their respective machines into the Teraflop range. This speed-up, evidently, will be realized both through new chip technologies as well as (and mainly) through increased processor numbers. Because of the latter aspect the first Teraflop machines will be prohibitively expensive and only very few installations, i.e., much fewer than the present numbers of Gigaflop performers, can be expected to be available to Computational Scientists.

A second feature shown in Table 1 is the considerable gap between peak and sustained performance. This gap is small for the 'mature' Cray for which optimizing compilers and considerable programming experience exists. In case of the Transputer-based machines the small gap is due to the scalar character of the FPU of the Transputer T800. The vector processing characteristics of the FPU on the i860 and a bottleneck regarding memory access makes it more difficult, even for a single processor, to achieve a sustained performance close to the peak value. In case of the present version of the CM-2 the limitation is due to the fact that the FPU's (Weitek) actually serve 32 processors and transfer of floating point data between memory and FPU's is less than optimal. Table 1 indicates then that the gap between peak performance and sustained performance is very much a matter of efficient programming of single processors. This is certainly the case, however, the data hide the fact that good performance of massively parallel machines working on a specific problem can only be achieved if the corresponding algorithms exploit the machines parallelism to the fullest extent. In fact, performance data for parallel machines are given for algorithms which work perfect in parallel, but most likely have nothing in common with the algorithms required for a problem of interest. A computational scientist who approaches a parallel computer with a given problem and with a program written for conventonal serial machines must expect to achieve much poorer performance, to the extreme that only a single processor among thousands of processors works on his problem. Only the Computational Scientist literate in parallel programming concepts and algorithms and willing to invest considerable effort in developing new programs can expect to harness the enourmeous increase in power which massively parallel computers promise.

In this respect, the problem of optimal employment of parallel machines appears to be compounded by the fact that the machines available today differ in the programming model they support, i.e., SIMD and MIMD machines with distributed and shared memory and with different data paths between processors. Fortunately, it appears that the parallel machines develop towards a more common ground, namely supporting mainly the MIMD concept through relatively coarse-grained architectures involving 64 bit processors with 4–16 Mbyte DRAM and also using hierarchical nets and

efficient routing schemes. The latter features imply that programmers need to differentiate memory only in two classes, fast on-processor and slower off-processor memory, the difference in access times for the latter memory residing on various parts of the machine being relatively small. As long as the band width of the net linking processors is not challenged, e.g., through the 'wicked' task of transposing a matrix across the whole machine, the future parallel computers will behave very close to shared memory machines, even though memory is physically distributed. Of course, Computer Science and manufacturers will support the user through various tools, well developed programming environments on hosts machines, vectorizing and, as far as this is possible, parallelizing compilers, debuggers and performance monitors.
However, to exploit the speed-up furnished by massively parallel machines, Computational Scientists cannot rely solely on such tools, but must spend intense programming efforts to fully exploit machine capacities. In my lecture I will describe such efforts in three areas of `Computational Biology`, namely `Structural Biology`, `Computational Neural Science` and `Magetic Resonance Imaging`.

Supercomputer Perfomance (in GFLOPS) 1991/1992					
1991			1992 (expected values)		
machine	peak	sustained	machine	peak	sustained
Cray Y-MP (4 processors)	2.7	1–2	Cray Y-MP (16 processors)	12	6–8
SC-400 Parsytec (400 processors)	1	1	GC-2 Parsytec (~256 processors)	6	5
Delta Touchstone (512 processors)	32	3–5	Sigma Touchstone (~2000 processors)	100–200	10–20
CM-2 (32,000 processors)	13.6	2–3	CM-5		

At this point it might also be mentioned that the optimism expressed above about the impending rapid development of computational resources is shared by many, in particular, also by Science Policy makers. Presently, the US (see, e.g., *Grand Challenges: High Performance Computation and Communication Initiative*, Report by the Committee on Physical, Mathematical and Engineering Sciences, National Science Foundation) and the Commission of European Communities (see, e.g., *Report of the EEC Working Group on High-Performance Computing*, Geneva, 1990) plan a large increase in funding of computational equipment, much of it for massively parallel machines, e.g., Teraflop computers. `Computational Science` needs to prepare itself to take advantage of the opportunities developing. This will require familiarity with concurrent computing and will require also investments in machines which are small versions of large production engines. For example, presently the fastest available computer for U.S. researchers is the 512 processor `Delta Touchstone` (see table above) at CalTech, whose smaller cousin, the `Gamma Touchstone` serves at various sites for code development and testing. Similarly, the next generation Connection Machine, the present CM-200 version of which matches the `Delta Touchstone` closely, is likely to be available in versions of different size. In Europe Parsytec plans a Teraflop computer based on the Transputer T9000 processor which, in keeping with the general concept behind the Transputer, will also be available at very different scales.

Opportunities Through Concurrent Computation Across Networks

The development of the high performance hardware described above can distract from the extremely important fact that massively parallel computation can be realized also on machines emerging from the low performance sector of computer technology. In fact, the advent of massively parallel computers coincides with the proliferation and rapid improvement of low cost computer workstations. Today

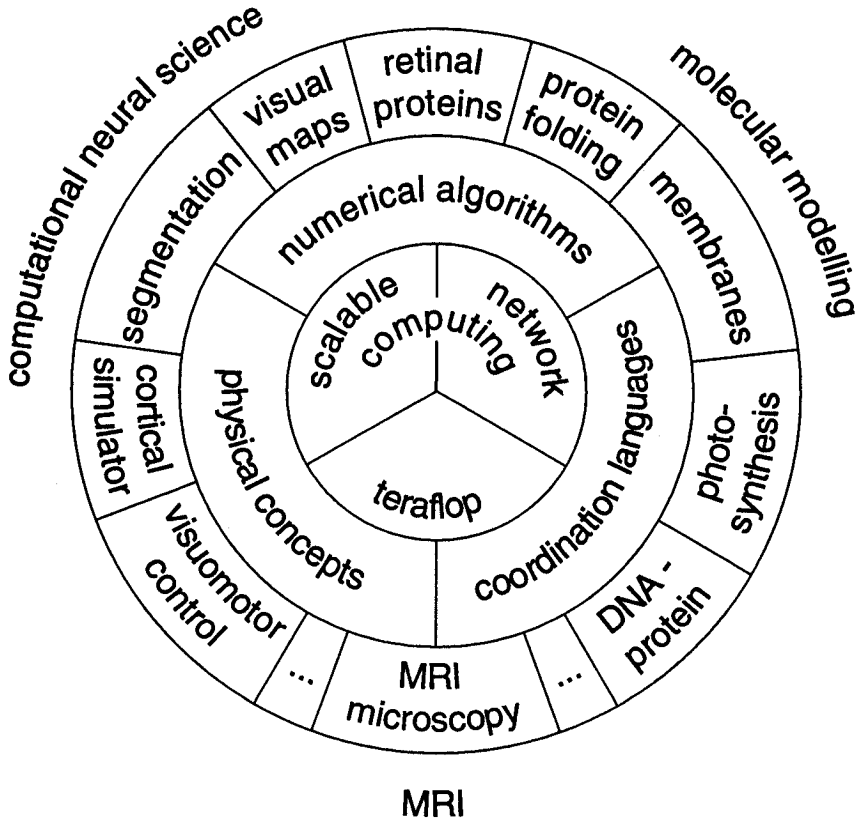

Figure 1: Schematic view of the activities of the Theoretical Biophysics Group at the Beckman Institute of the University of Illinois. The core of the diagram denotes the hardware used, the first layer the algorithm development and the outer layer the various applications for massively parallel computing pursued by the group.

a desktop workstation with 2 Mflops performance costs only about $5,000 (e.g., a NeXTstation) and can easily be configured within a wide network. *Coordination languages* for organizing computation across a massively parallel computer can be employed also over a network, and tasks which require only moderate message passing between processors can obtain Cray YMP processor speed through about 100 such workstations. This estimate suggests that enourmeous computational resources can be harvested from existing computational equipment. It is also apparent that concurrent computing across networks opens interesting alternatives to installations of single, large scale computers. Again, this capacity will only be available to computational scientists if efforts are spent on adapting algorithms to concurrent computation and to coordination languages.

2 Concurrent Computation in Biology and Medicine – An Overview

Figure 1 presents an overview of the author's efforts to employ parallel computation to solve problems in Biology and Medicine. The efforts can be described as structured in two layers surrounding a core. The core consists of three sectors, corresponding to the three different types of hardware employed. The inner layer describes algorithm development and the outer layer the various biological and

medical areas addressed by means of concurrent computation.

Hardware

Let us begin describing the hardware component of Figure 1. The sector "teraflop" symbolizes very large scale parallel machines, i.e., presently a 32K processor Connection Machine CM-2 and a 512 processor Touchstone Delta, which are used by the author's group for research in Structural Biology and Computational Neural Science. Such machines are available at National Centers and computer time is available either through competitive proposals or through contractual agreements. The National Centers are expected to furnish during the next years the fastest and largests machines available to be used for the most demanding and most relevant Science and Engineering projects. Presently, the author's group uses machines at National Centers to describe the reaction mechanism of proteins, in particular, those of the class of visual receptors, and to simulate the formation of so-called brain maps, in particular, the representation of visual images in the optical cortex. The machines are programmed in C and parallel extension of this language.
The second core sector, "scalable computing", denotes parallel computers which are scalable, essentially from single processor to multi-processor machines. Presently, the author's group operates in this category Transputer-based machines of various types: a self-designed and self-built 60 processor machine which began operation (as a 12 processor version) in August 1988, an 18 processor twin machine, and commercially available VME-bus boards with 4 and 6 processors, respectively, connected to a Sun-4 and Silicon Graphics workstation. These machines are programmed in Occam II as well as in C (Par.C). The machines are used for very large scale simulation of biopolymers, e.g., water–DNA–drug systems, membrane–protein systems and protein complexes. For such computations the 60 processor machine achieves the speed of about a single Cray 2 processor as judged by comparision with commercial programs (Charmm of Polygen, Inc.) running on the Cray 2. The Transputer-based machines are also employed for simulations of Magnetic Resonance Imaging, in particular, regarding the application of this method to microscopy with about 10 μm resolution.
The third core sector "network computing" describes the use of Sun workstations for parallel computation employing the "Linda" coordination language. This system was used to simulate medium-sized proteins. Presently, the author's group extends these calculations to networked Silicon Graphics workstations and to networked NeXT workstations.

Benchmarks

Benchmarks of the various computers employed are provided in Fig. 2. On all computers has been carried out a molecular dynamics simulation of the protein lysozyme. The benchmark test measured how many integration steps, each 0.5 fs long, were performed on the various machines. The calculations employed the program MD written in C, except were stated (Xplor written in Fortran on the Cray 2, EGO written in Occam II on the Transputer-based machine). The performance has been measured relative to that of MD running on the Silicon Graphics 320 VGX.
The test calculations showed that 60 Transputers T800 can match about the performance of a Cray 2 processor, demonstrating the suitability and the good cost/performance ratio of the Transputer-based machine (parts cost of our machine with 60 processors and 240 Mbyte DRAM were at 1988, 1989 prices $60,000). One should keep in mind that the Transputer is very scalable and a machine like ours, which is available from vendors in similar configurations, can grow incrementally according to need and budgetary possibilities.
The most exciting entry in Fig. 2 is the (estimated, explicit runs have been completed sofar only on a network of Sun workstations) performance of a network of 16 NeXT workstations. Such network

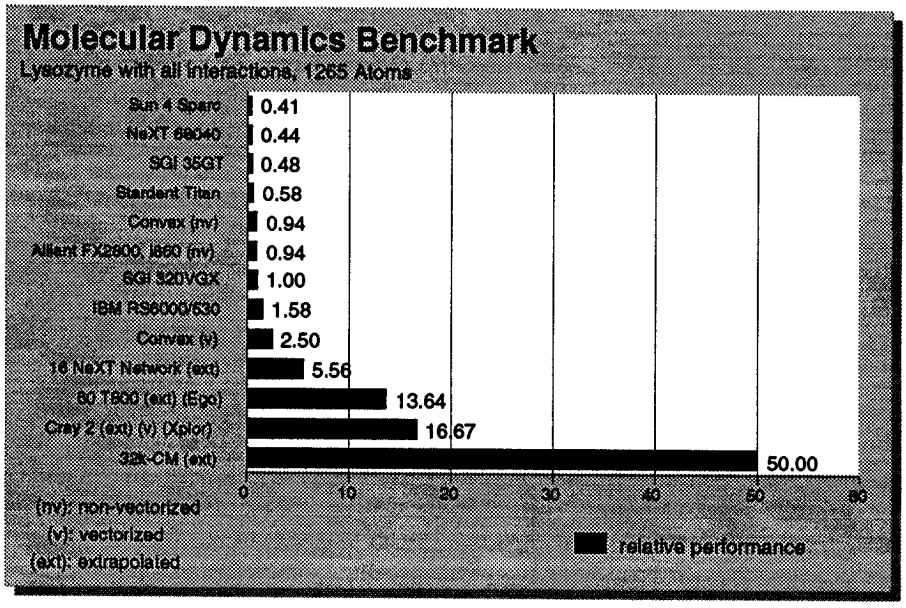

Figure 2: Performance on a number of different platforms. All calculations were made using the program MD, except those made with X-PLOR on the Cray-2 and EGO on the Transputer system. The result for the NeXT-network was extrapolated assuming an 80% efficiency. The CM-2 performance was extrapolated from an 8 K-configuration because of the small size of the molecule lysozyme. For larger molecules the CM-2 performance would be even better than indicated. Additionally, an improvement by a factor of 3–5 can be achieved on the CM-200 by taking advantage of the new slice-wise architecture. (Figure provided by Andreas Windemuth)

is available for under $100,000 and obviously provides a very flexible, multi-purpose solution with a good pice/performance ratio, in particular, in case that existing networks can be utilized.
Another interesting entry is the Connection Machine CM-2 with 32,000 procesors. The tests show that this machine can be utilized well for the simulations. In this respect it should be pointed out that the program, in this case, actually runs on the Sun front end of the Connection Machine, only the (most time consuming) evaluation of non-bonding pair forces being evaluated on the CM-2.

3 Projects

We continue the discussion of Fig. 1, namely, of the outer layer describing various Computational Biology projects being carried out on the various parallel machines mentioned above. In the lecture only a subset of these projects will be described in some detail. These projects are connected with the study of biological vision.

Molecular Dynamics of a Retinal Protein

Retinal proteins act as the receptors of light in the eyes of all animals. The protein resides in a membrane of a receptor cell and through absorption of light is very rapidly switched to a state which, after a series of biochemical amplificaton steps, alters the intracellular potential of the receptor cell and, thereby, transmits an electrical signal to the brain. We have actually investigated a retinal protein

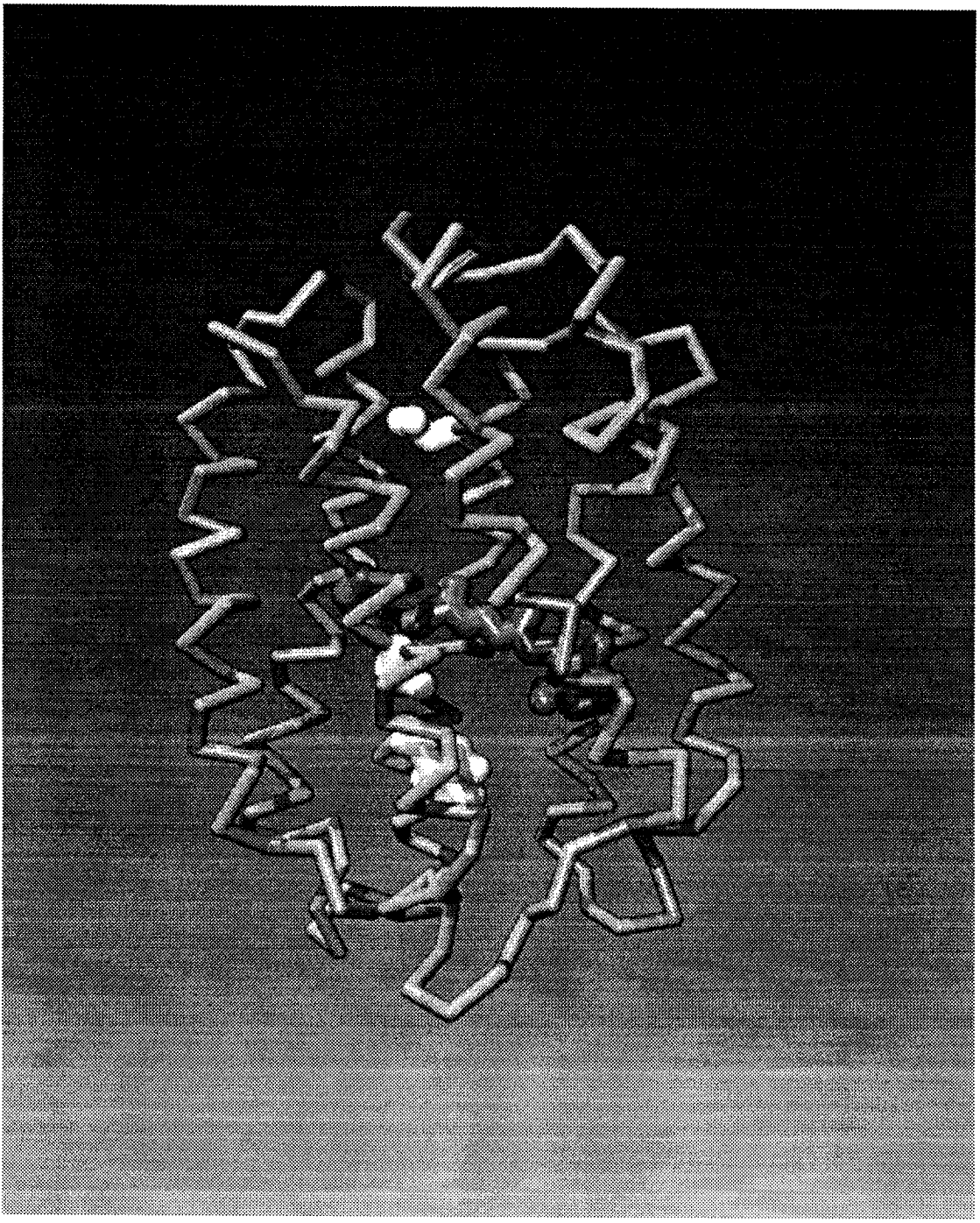

Figure 3: Schematic view of the protein bacteriorhodopsin. Shown is the backbone of peptide bonds together with some important amino acid side groups and with the prosthetic group retinal. Retinal (the large molecular moiety in the center of the protein) absorbs sun light and transforms itself within about 10^{-13} s into a new isomer. The back-reaction of retinal to the initial isomeric state is coupled to transfer of a protein from one side (top) of the protein to the other side (bottom) generating, thereby, a cellular electrical potential. The simulatons on the Connection machine CM-2 completed and refined the structure of the protein as well as identified the nature of the initial photoreaction. (Figure courtesy of A. Windemuth)

in a bacterial cell, namely bacteriorhodopsin, simulating the very fast reaction triggered by light and the subsequent thermal reaction steps. The novelty of the investigation, which utilized our program MD running on the Connection machine CM-2, has been that the structure of the protein had been available only partially and at poor (3.5Å) resolution. Hence, we needed to complete the structure and improve structural defects. Such efforts in structure completion and structure refinement will become commonplace in the near future, in particular, in connection with structure determination through 2-dimensional NMR spectra. Such calculations require very large computational resources as provided by massively parallel computers or through concurrent computation across networks. In the lecture I will also briefly describe molecular dynamics simulations of biological membranes

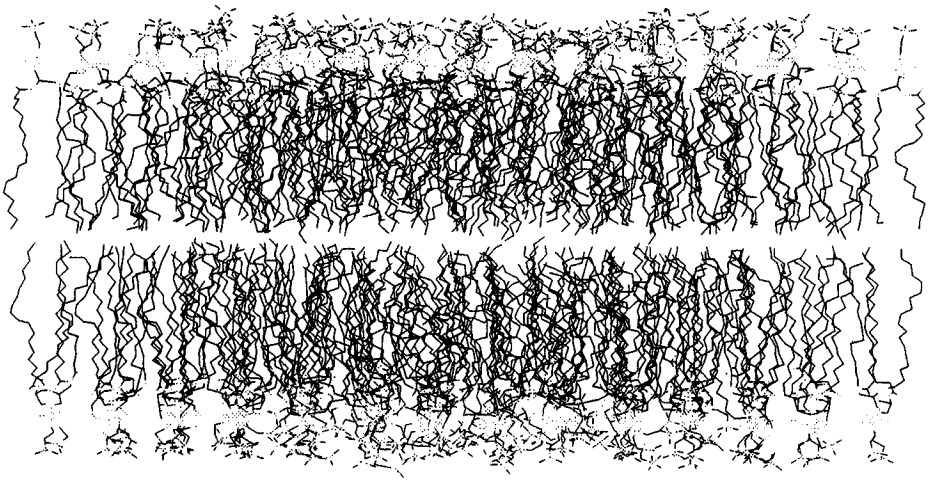

Figure 4: View of the lipid bilayer from the side. The structure has been equilibrated for about 20 ps. (Figure courtesy of H. Heller)

and of the interaction of drugs with DNA. These latter simulatons were carried out were carried out on our self-built Transputer-based computers. Figure 4 shows the structure of water and lipids assumed by a lipid bilayer after about 20 ps of simulation. The figure mainly documents that the Transputer-based machine is capable of simulating systems made up of 25,000 atoms.

Visual Maps in the Optical Cortex

In the lecture we will also report on a second project which is concerned with the representation of visual images in the brain. Such representation is achieved through an ordered connection between cells of the brain's visual cortex and receptor cells of the retinas of the two eyes. Through a technique, called 'voltage sensitive dye technique' much detail is known about this representation from laboratory investigations of monkeys and cats. In fact, one knows well the filter function (receptive fields) of cells in a whole array of the visual cortex, the brain area concerned with primary vision. The distribution of receptive field properties is called the 'visual map'. Most interestingly, these maps are established during the postnatal phase of an animal's life and development is driven through visual experience, i.e., the maps are individually 'learned'. We have studied the rules which underly this learning process.

A few numbers are relevant here. A brain area of a few mm^2 entails about 10^5 neurons which each have about 1,000 connections to the retinas of the two eyes, i.e., a 36 mm^2 patch of the brain (see Fig. 5) is endowed with about 10^9 connections (synapses). Actually a modification of these synapses is the basis of the formation of the visual maps. To simulate the map formation one needs to deal then with dynamical systems of about 10^9 time-dependent variables. Such investigations, presently,

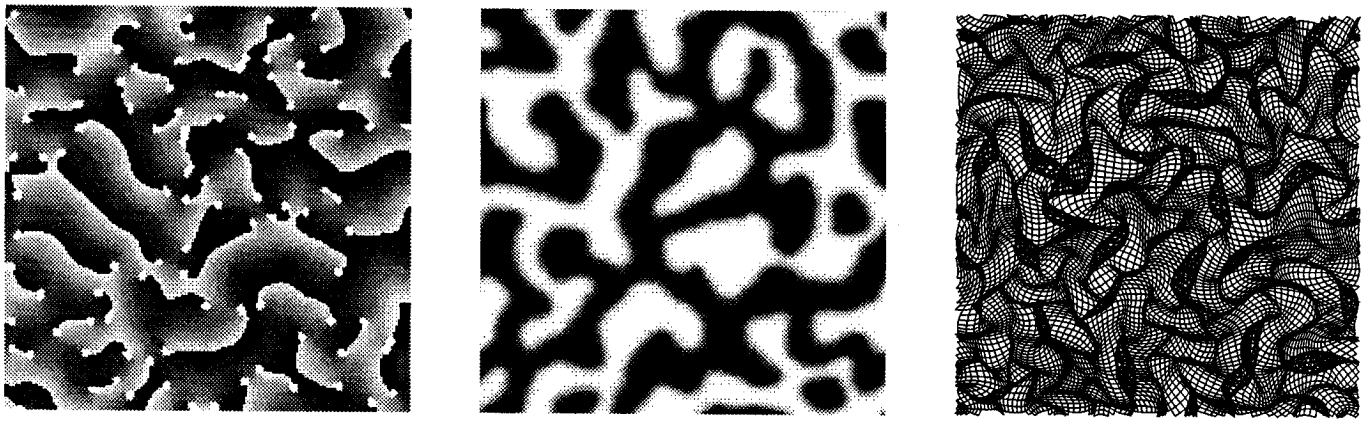

Figure 5: Visual maps simulated on the Connection machine CM-2: orientation preference (left), ocular diminance (center) and locations of receptive field centers (right) in a small area (about 36 mm^2 of the visual cortex). (Figure courtesy of K. Obermayer)

can only be carried out routinely on massively parallel machines. We have employed for this purpose the Connection Machine CM-2. Figure 5 shows a mature visual map as generated in a simulation on the CM-2. The simulation run at a computing level of 2.5 Gflops which lies well within the range of optimal sustained performances of a 32,000 processor CM-2 (cf. Table 1).
The visual map in Fig. 5 (left) presents , the so-called orientation preference map, which codes in shades of gray, the orientation sensitivity of the cortical cells, covering the orientation interval [0^o, 180^o]. Since black codes for 180^o orientation and white codes for 0^o orientation, the orientation around these values appear discontinuous in the map, but actually are continuous. The white points in the map correspond to zones where brain cells are actually insensitive to orientation. Near these zones the visual cortex stablishes cells which are sensitive to texture (granularity of the surfaces in the field of view) and sensitive to color.
Figure 5 (center) presents the same neurons as on the left hand side of the figure showing, however, the sensitivity to input either from the left (black) or right (white) eye, the so-called ocular dominance. This part of the figure, the so-called ocularity map, demonstrates that the cells are also transmitting stereo information to the brain. The right hand side of Fig. 5 presents the so-called retinotopic map for the same cells as the two other parts of the figure. The figure shows how the mesh of a net seen by one of the eyes would be represented in the brain (actually, the figure shows the inverse of the situation just described). One can recognize that such net is orderly represented except for some wrinkles. These wrinkles are unavoidable since the map represents also orientation information. One can show, however, that in some sense the brain tries to keep the wrinkles of the mesh at a minimum. All three parts of Fig. 5 taken together explain then how visual images are actually presented in the brain: the brain processes the retinal images on its way to the brain such that several attributes are compressed into a single brain representation: (1) position in the field of view, (2) orientation of local edges, (3) ocularity, (4) texture, and (5) color. Another important result of the simulation, which we cannot explain here in any detail, is that a small set of rather simple principle suffices to generate the visual maps through visual experience of an individual animal.

Visuo-Motor Control of an Industrial Robot

The brain is an organ the ultimate function of which is to allow an organism to react to sensory information. We have shown in our work that the principles involved in the development of visual

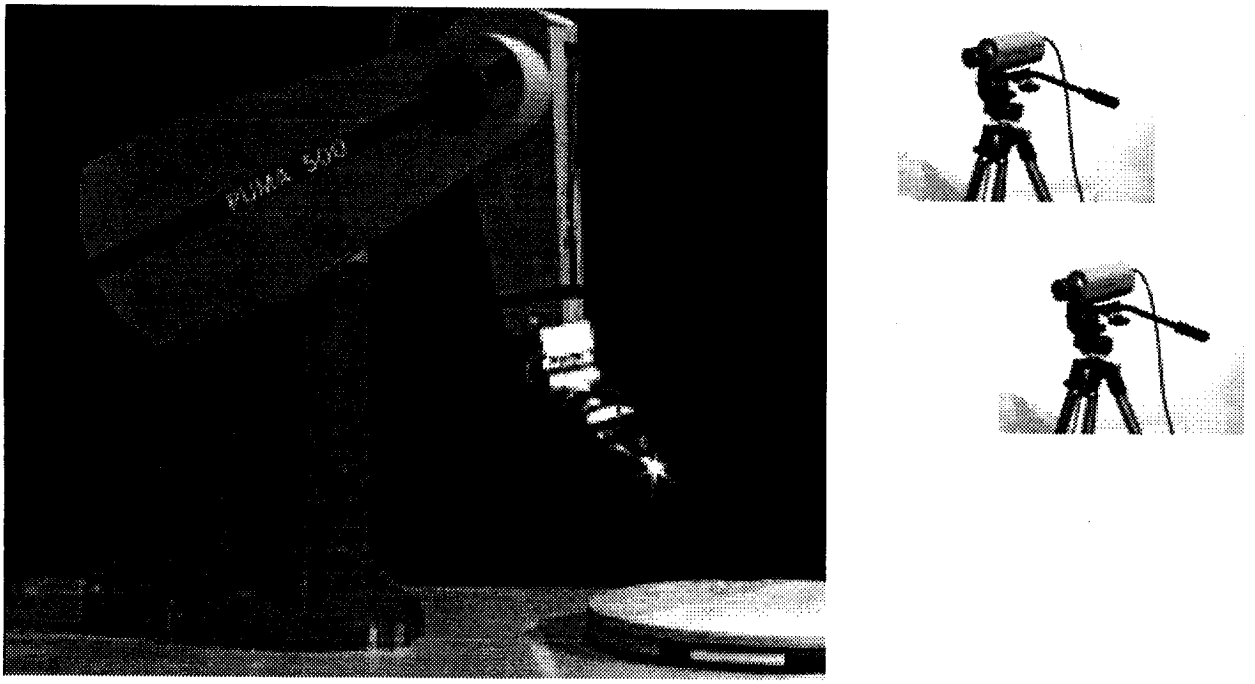

Figure 6: Presentation of the robot–camera system which, endowed with a computer program simulating the principles of neural developmental processes in biological organisms, learns to movethe robot's end effector (light blinking at the end of the robot arm) to any place in the work space. (Figure courtesy of J. Walter)

maps in young animals can be applied also to other capacities of the brain, for example, to motion control. In order to demonstrate that the principles assumed and translated into computer programs provide a basis for the acquisition of proper motion control in an animal, we have employed our programs to a system of two stereo cameras and an industrial robot arm. Such system is shown in Fig. 6. We have demonstrated that the developmental principles postulated by us can serve to teach a robot–camera system to move properly and accurately. For example, the industrial robot Puma 560 can learn after about 2,000 trial movements to point its end effector with a high degree of precion anywhere in the robot's work space. This is achieved by the system through learning first to present the work space in an orderly fashion in a neural net structure which exists as a data structure in a Sun 4 driving the system. The system can then recognize visually its errors in moving the end effector and through a simple learning rule adapts itself to move properly. The map which connects the back planes of the camera to the neural network presented in the computer has many similarities to the map presented in Fig. 5. The signals which drive the motors at the joint of the robot arm are presented as simple vectorial and tensorial data stored at the nodes of the neural network structure.

4 References

Some of the material presented in the lecture can be found in the forcecoming textbook *Neural Computation and Self-Organizing Maps: An Introduction* by Th. Martinetz, H. Ritter, and K. Schulten (Addison&Wesley, New York, 1991). Other publications covering material mentioned in the lecture can be obtained from the author upon request.

[0] **Textbook:** *Neural Computation and Self-Organizing Maps: An Introduction*, H. Ritter, Th. Martinetz, and K. Schulten (revised, English edition, Addison-Wesley, New York, 1991)

[1] **Development and Spatial Structure of Cortical Feature Maps: A Model Study**, K.Obermayer, H.Ritter, and Schulten in *Advances in Neural Information Processing Systems 3*, D.Touretzky and R.Lippman, Eds. (Morgan Kaufmann Publ., New York, 1991)