Imagine a computer that expands your vision

by looking at the whole problem at once.

Imagine a 1000 Mips computer that is simple to program. That's right. 1000 million instructions per second. And simple to program.

A computer that solves your problems by working on all of the data at once, instead of one word at a time.

A computer that handles words and pictures as easily as numbers.

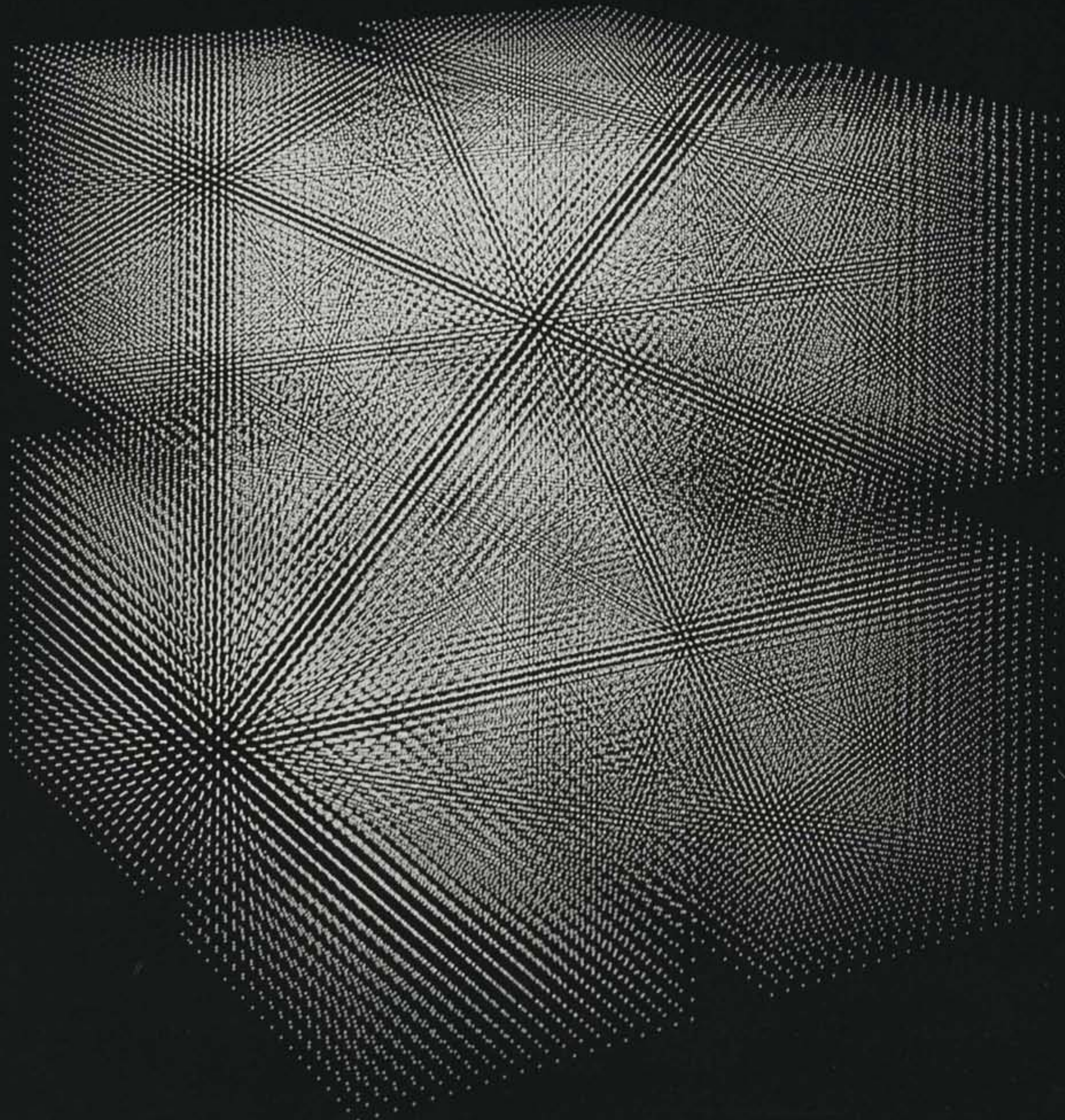
A computer that is opening new frontiers in applications.

The Connection Machine® computer system is here. Reversing a twenty-year trend toward greater and greater complexity in large scale computing systems. Uniting immense processing power with simplicity of programming. Establishing a new standard of large scale computing capability.

The Connection Machine system is easy to use because it has enough computing elements to assign one to each piece of data. Its 64,000 processors scan whole data bases, process whole pictorial images, and simulate whole VLSI circuits simultaneously. All processors are linked together in a dynamic network that reconfigures to match the way the problem works. The result is a simpler to program computer, one that delivers its enormous computing power more naturally and directly.

What do we mean by "simple to program"? We mean that many of your most complicated and time-consuming subroutines can be replaced by single operations that are executed directly by the machine. Not just regular operations like vector arithmetic, but also operations that require complex patterns of data transfer, like network propagation and sorting.

We also mean that it is easy to use. You interact with the familiar programming environment of a front end system, running the editors, utilities, and operating system to which you are already accustomed. You write your program in a simple extension of a high level programming language, like C or Lisp. The compiler and the Connection Machine hardware automatically take care of how the data is allocated to the processors, how operations are synchronized, and what data is communicated from one processor to another. It's good at those things, so you can concentrate on the application, not the computer.



Looking at the whole problem at once means computing on every element of a data structure at the same time. The Connection Machine system's tens of thousands of processors allow it to do this in a straightforward way: by attaching a separate processor to each element of a data structure.

The type of data structure depends on the application. In many language processing applications, data level parallelism means a processor for every word or every meaning. For data base applications, it means a processor for every document. In a numeric simulation, it may mean a processor for every element of a matrix. And in image processing, data level parallelism means a processor for every picture element, or pixel, in an image.

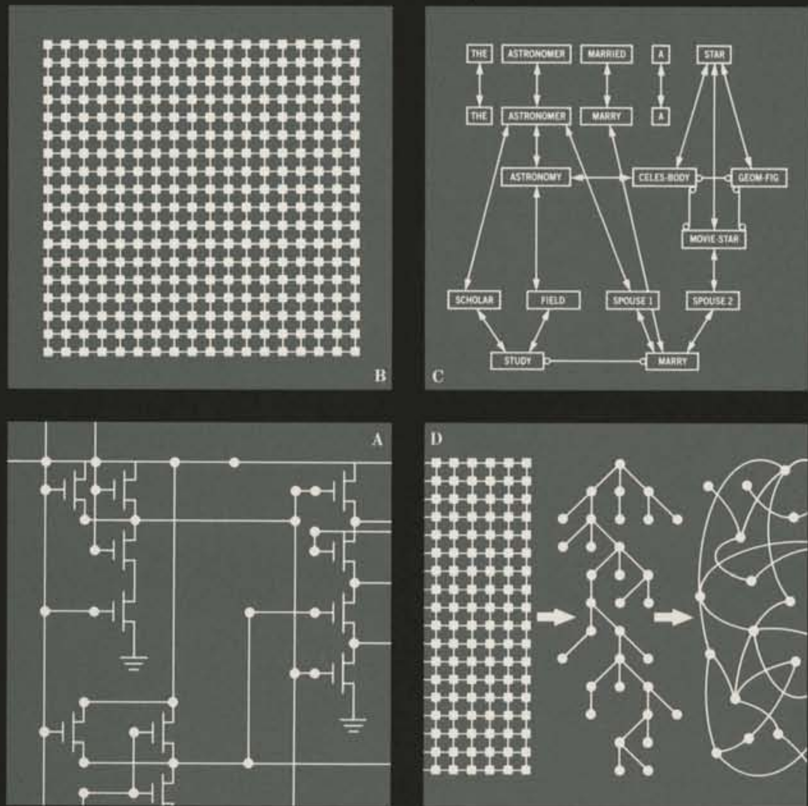
The performance advantages of data level parallelism are dramatic. The parallelism inherent in the data structures grows with the size of the data. Processing all the pixels in an image, for example, is as fast as processing one pixel, because they can all be computed at the same time. A whole data base can be searched in the time it takes to search one document. When computing the flow of air over an airplane wing, the Connection Machine system calculates the flow over all parts of the wing simultaneously, just like it works in reality. Speedups of 1000 or more are common with data level parallelism.

If data level parallelism is so simple and powerful, why don't all big computers work this way? Because in order to make these programs work, tens of thousands of processors have to work together. They have to communicate. The Connection Machine system solves the communications problem in hardware, so the natural algorithms work well.

What makes the Connection Machine system work are the connections. Inside the machine there is a very flexible high-bandwidth communication network that moves data between processors at billions of bits per second. Routing circuits on every chip automatically steer data along the fastest paths, helping to make programming simpler. There is no need to adapt your application to the structure of a fixed architecture like a grid, ring, hypercube, or tree. Instead, the Connection Machine system adapts to your application, by dynamically forming the connections that are needed.

(A) In the simulation of a VLSI integrated circuit, the individual problem element is the transistor. Transistors connect together by their circuit wiring, which is unique to the individual circuit. Some transistors are wired to the transistor next door, some are wired to transistors clear across the chip, and others are wired to both. For every VLSI circuit it simulates, the Connection Machine system reconfigures its internal processor connections to match the wiring of the circuit.

(B) In picture processing, the individual problem element is the pixel. Pixels interact in a grid pattern, with the state of each pixel influencing those adjacent to it. For image processing, the Connection Machine system configures itself as a grid, allowing each processor to pass information to the processors above, below, and to the sides.



(C) In language processing, the individual problem element is the word or the meaning. Words and meanings connect together in increasingly complex structures called semantic networks. The Connection Machine system sets up a connection between each related word and meaning.

(D) The most demanding applications have no fixed topology. They change during the course of the problem. So does the Connection Machine system. A complete vision system may, for example, start its analysis by filtering the image in a grid topology. Global operations, such as computing the average intensity, are naturally tree-like in nature. Bringing together features into objects requires irregular patterns of communication. Only the Connection Machine system adapts dynamically to these changing patterns.

In short, the Connection Machine system's simplicity is also the source of its great power. Large numbers of processors plus a dynamically reconfigurable intercommunications network combine to allow processing to go on directly at the data level, where thousand-fold parallelism is natural.

Connection Machine simplicity and ease of programming extend across the full range of applications where the amount of data is large. In numeric and symbolic applications alike, the system's ability to dynamically configure itself to the natural structure of each problem makes algorithms more natural and implementation more rapid.

Example Application: Words. The Connection Machine system is allowing certain artificial intelligence algorithms to be applied to real-world problems for the first time. One of the most important of these applications is data base search.

The Connection Machine data base system retrieves information by whole document comparison instead of by individual "key word." The user can point to a single relevant document and say, in effect, "Find me all the documents on the same subject as this one." The system compares content-bearing words and phrases throughout the documents. Up to 64,000 whole document comparisons are performed simultaneously.

Whole document comparisons consistently find relevant documents that keyword-based systems miss, because they do not depend on matching any single word. Within a document there are scores of content-bearing words. Many will appear in other documents on the same subject, assuring that an overall match is found even though particular words may be missing. The second advantage of whole document comparison is that it can present the best documents first. Documents with the most significant matches are selected for display at the top of the list.

The Connection Machine retrieval system also eliminates manual indexing of unstructured text. Even huge volumes of incoming text are analyzed automatically by Thinking Machines indexing algorithms, which pick out the content-bearing words and phrases. These phrases form the basis for the subsequent search decisions.

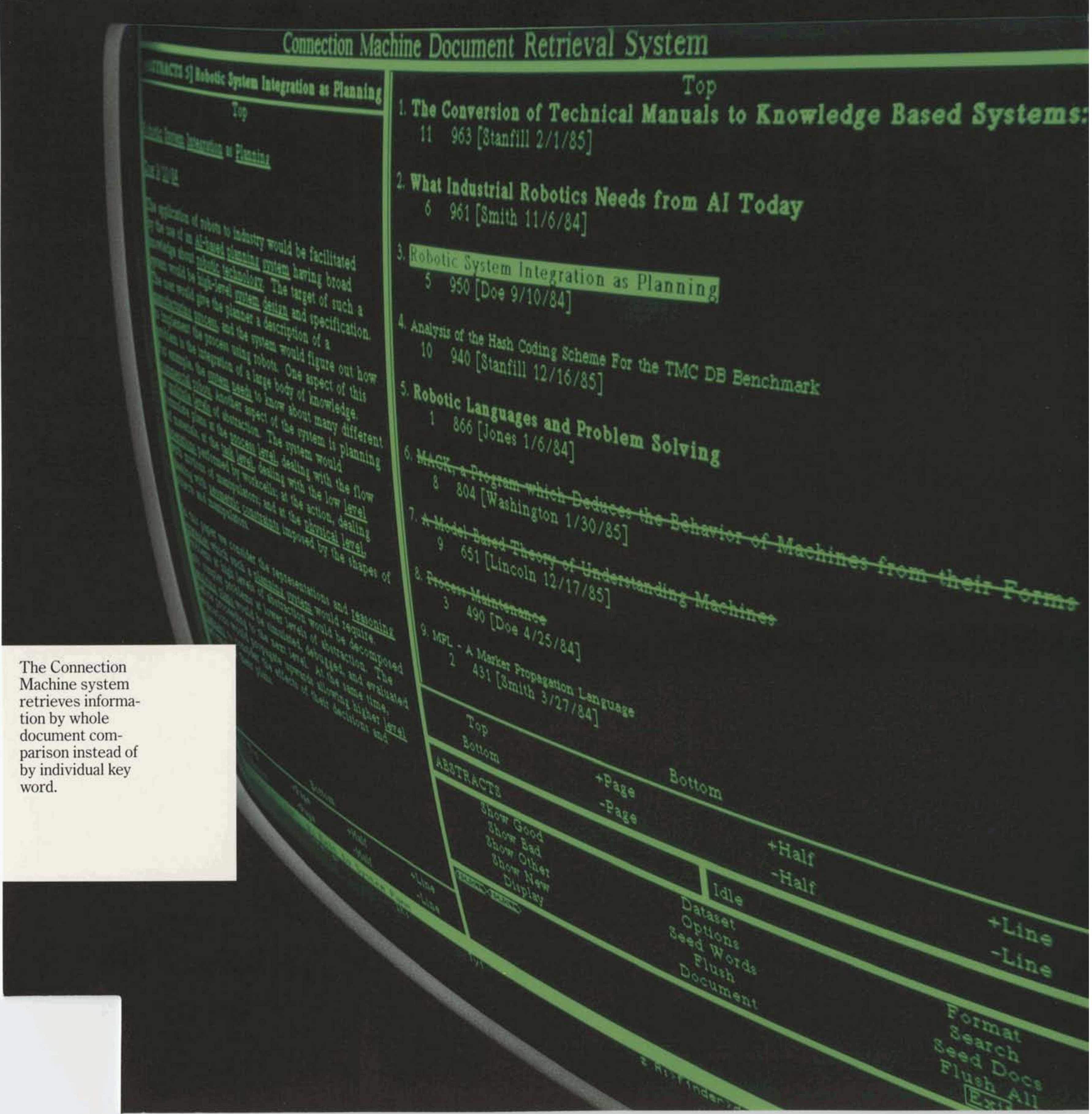
"Whole document" algorithms are just one of the areas of natural language computing that the Connection Machine system is making easy to implement. Interestingly, many of these approaches have been known for 20 years, but until now they have been too difficult to program.

Example Application: Pictures. Today, to gain useful information from satellite images, users must analyze pictorial information directly, and in very high volume. Modern analysis of food production, land terrain, and weather patterns all depend on these pictures.

The Connection Machine system has made it possible to automatically generate contour maps in a few seconds instead of a few hours. The algorithms to do this utilize two images of the same terrain taken from slightly different angles.

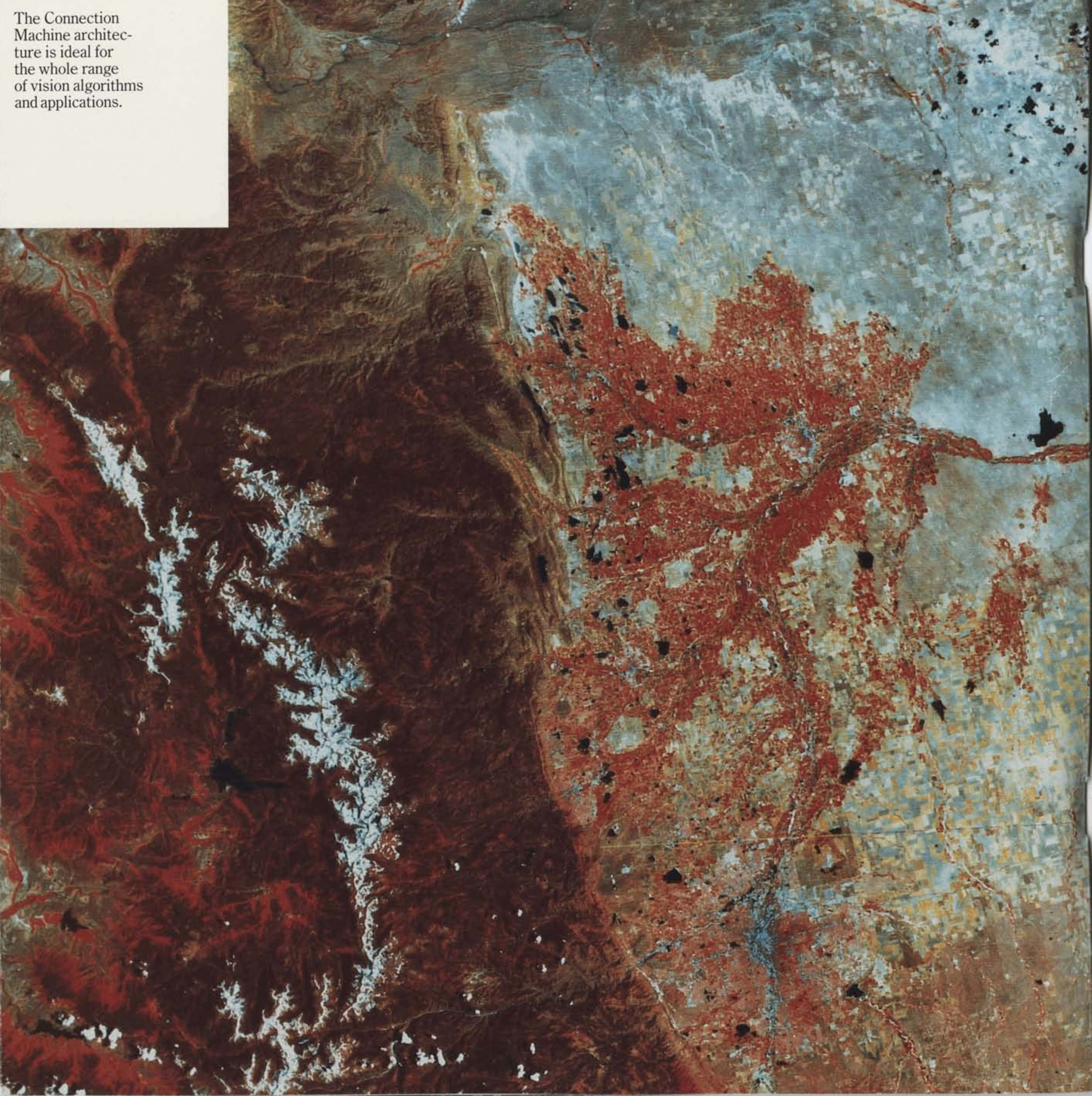
After the noise in the images is removed, they are "slid" over each other and minute features compared. Thinking Machines comparison algorithms note the alignment of features at each stage. The more a feature is displaced between the two images, the closer it is to the camera and hence the higher it is in physical terrain terms. This information allows the system to draw and display a contour map of the terrain.

The Connection Machine architecture is ideal for the whole range of vision algorithms and applications. Region labeling algorithms, for example, pick out areas of consistent intensity within a picture and mark them as single objects. Other algorithms pick out additional significant features, such as lines and arcs, for further analysis. The combination of these capabilities, many of which operate at a thousand times the speed of conventional computers, makes the Connection Machine system the computer of choice for demanding vision applications.

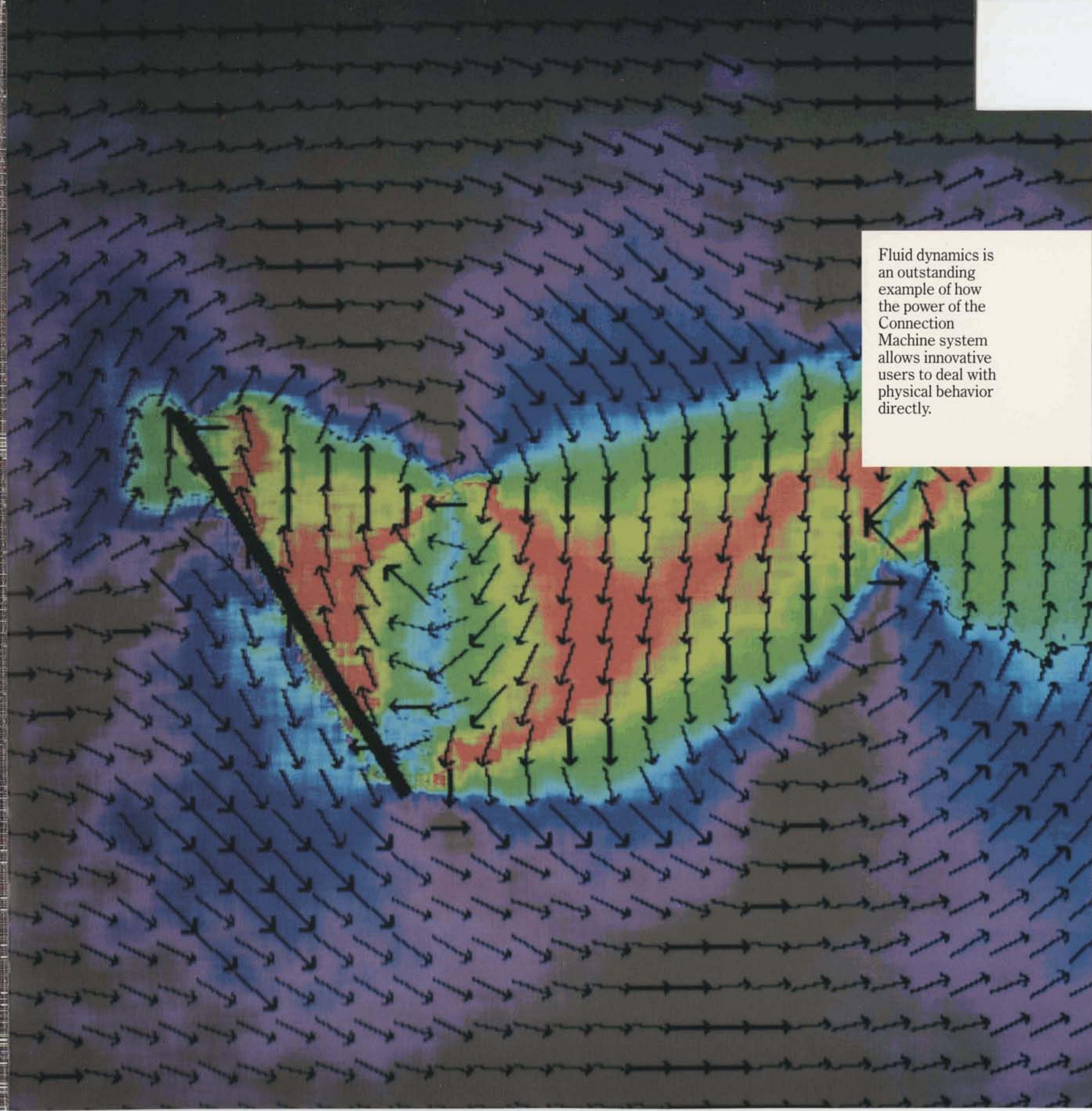


The Connection Machine system retrieves information by whole document comparison instead of by individual key word.

The Connection Machine architecture is ideal for the whole range of vision algorithms and applications.



VLSI simulation is just one of the two-dimensional and three-dimensional numeric problems the Connection Machine system is solving.



Fluid dynamics is an outstanding example of how the power of the Connection Machine system allows innovative users to deal with physical behavior directly.

Example Application: Numbers. The Connection Machine system brings a new level of power and naturalness to numeric computing applications. The simulation of a VLSI circuit shows how general communication and massive computing power combine to provide an easy-to-program solution. The problem is to feed a wave-form into a circuit, then compute the wave-form that the circuit will produce at any other selected point. For a circuit with tens of thousands of transistors, all switching on and off in parallel, the required number of computations is immense.

The Connection Machine system configures its communications pattern to match each individual circuit exactly. A complete processor is assigned to each transistor and node. The linkages between processors are configured to match the wires between transistors. The simulation matches the reality: all processors (transistors) compute (change state) in parallel.

Processors evaluate mathematical equations that describe the behavior of the device they are simulating. Then they communicate their voltage, conductance, current, and charge to other devices. Finally, the system iterates to reach consistent values for all devices for that time step of the simulation.

VLSI simulation is just one of the two-dimensional and three-dimensional numeric problems that the Connection Machine system is solving. Other examples include matrix algebra and linear programming.

Example Application: Aggregate Behavior. Powerful though they are, words, numbers, and pictures are simply abstractions of the world around us. They simplify behavior that is too complex to deal with directly.

Fluid dynamics is an outstanding example of how the power of the Connection Machine system allows innovative users to cut through these abstractions and reach the physical behavior directly. By analyzing the way fluids flow, scientists design more efficient aircraft and make more accurate weather forecasts. Traditional methods have modeled fluids by partial differential equations. The Connection Machine system provides new insights by more directly modeling the behavior of actual molecules.

Huge numbers of packets are introduced into the system, each like a tiny bundle of molecules. They move and jostle according to very simple logical rules. No arithmetic of any kind is involved, so the programming is simpler than traditional methods.

As these individual local interactions evolve in huge numbers (the Connection Machine system can update a billion of these individual states per second), the behavior of fluids emerges. Injected from an edge, a fluid flows in a regular way until an obstruction (such as an airfoil) is encountered. Swirls and eddies appear. Over time they lose their energy and trail off.

The simplicity of the model makes simulation an interactive process for the first time. A change in the geometry of the object requires no lengthy reinitialization. Results for a new shape are available in a minute or two.

Programming and Using the Connection Machine System. We no longer have to accept complexity as the price of increased performance in high end computers. The Connection Machine system achieves performance without tricky compilers, ultra-fast components, or exotic packaging.

Simple Construction. The Connection Machine system uses a few simple parts over and over again. A special processor chip implements sixteen processors. 4096 of these chips are used in the system as a whole. Design rules throughout the system are conservative. There are no exotic technologies to cause reliability problems. The system is entirely air cooled. No unusual site preparation is required.

If a failure does occur, fault diagnosis is rapid. Each processor needs only to check itself. 65,536 processors do this testing in parallel. Most failures are isolated to the chip level within five minutes. Hardware is repaired by replacing modules. There are only seven kinds of modules in the system, so a complete set of spares is easily maintained at every site.

Familiar Operating Environment. The user interacts with the Connection Machine system through a conventional front end system, such as a VAX[1] or a Symbolics 3600.[2] The front end supports the operating environment. Installation of a Connection Machine system does not mean introducing a new operating environment into your data center.

The system is programmed via the front end, using familiar editors and utilities. File structures and network protocols are supported there as well, as are the full range of standard VAX and Symbolics peripherals.

A Closer Look: Programming at the Data Level. Computers operate by having a program (the control sequence) operating on a set of data elements to solve a particular problem. Systems that use "control level parallelism" apply their processors to individual sections of the program. The Connection Machine system applies its parallelism directly to the data. The amount of parallelism that can be exploited grows with the size of the data. There is no need to segment the program.

If the data structures contain fewer than 65,536 elements apiece, the Connection Machine system makes the assignment directly. If there are more, the system operates in virtual processor mode, simulating a larger number of processors in a way that is transparent to the user's program. The system can easily support up to 1,000,000 processors in the virtual processor mode.

Variable word length programming and dynamic reconfiguration complete the task of matching the program's data exactly. Data in a Connection Machine system may be as small as one bit or as large as thousands of bits. For picture processing, one- to eight-bit values are common. For numeric processing, 16- to 64-bit words appear most frequently. Language processing values, such as words and sentences, can vary from a few bits to thousands. The Connection Machine system handles them all with equal efficiency.

Dynamic reconfiguration matches the data's connectivity as well as the data itself. When the data structures are set up, so are the linkages. Any processor can send data to any other processor. Within the Connection Machine language environment, data connections are carried out automatically. At the hardware level, the system supports interconnection with a communications system called the router. All 65,536 processors can exchange data simultaneously.

The C* Programming Language. C* is a direct implementation of the data level computing philosophy for general purpose computing. The language is so similar to C that it requires no separate compiler. C* is implemented simply as a preprocessor to the standard C compiler. Consistent with the tenets of data level parallelism, there are no changes to the basic command structure. The control sequence of a C program does not change within C*. Enhancements are at the data level, allowing data structures to be connected to individual processors for rapid execution.

The *Lisp Programming Language. *Lisp is a direct implementation of the data level philosophy for artificial intelligence applications. *Lisp adds data structures to allow direct control of memory allocation and assignment of data values to processors. It gives the programmer direct access to, and control over, the Connection Machine hardware, but does so from within the Lisp programming language. Thus the *Lisp user retains all the productivity benefits of the Lisp machine environment.

The Connection Machine system uses a few simple parts over and again. A special processor chip implements sixteen processors. 4096 of these chips are used in the system as a whole.

The Future Belongs to Computers That Look At the Whole Problem At Once.

The Connection Machine system operates on a problem's entire data set at once. It is a strikingly simpler approach that opens up whole new possibilities for problem-solving. For picture and image processing, it provides performance levels thousands of times greater than conventional machines. For language processing, it provides access to information in the way people can most easily use it. For scientific processing, it provides access to physical phenomena never before captured in computer simulation.

Yet the impact of data level computing is only just beginning. As innovative scientists are becoming conversant with the Connection Machine system's potential, they are inventing new algorithms and approaches to problem-solving. They are merging the image, language, and numeric processing capabilities of the system into higher level algorithms. Thinking Machines Corporation itself is at the forefront of this new wave of computing. By using knowledge in all its forms, we see the potential for computing systems far more useful and far more powerful than anything available to us today.

Some day we will
build a thinking machine.

It will be a truly
intelligent machine.

One that can see and
hear and speak.

A machine that will
be proud of us.

Thinking Machines Corporation
245 First Street
Cambridge, Massachusetts 02142
(617) 876-1111