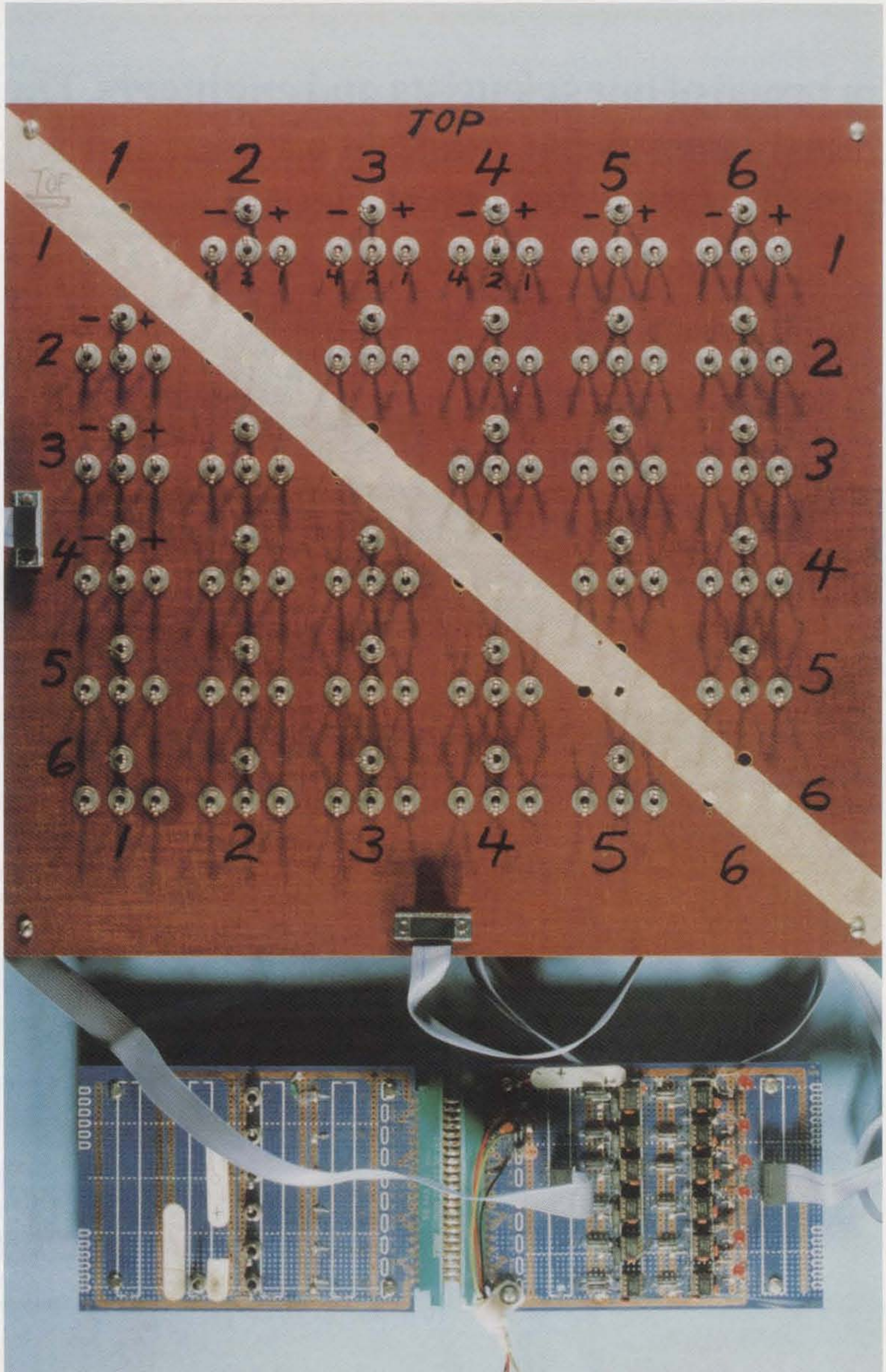


TOP
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6

Neural Networks: The Caltech – JPL Connection

The world's first man-made feedback neural net had six fully interconnected neurons. Each connection used a group of four toggle switches—the upper one chose whether the connection was inhibitory (−) or excitatory (+), and the lower three set the connection's strength. The six toggle switches on the left side of the blue circuit board input a pattern, and the column of light-emitting diodes (LEDs) to the right lit up with the output.

Imagine an autocratic chef, who can't stand to have another presence in the kitchen, preparing a seven-course banquet. Taped to the wall is a master instruction list that combines all the individual steps from all the recipes on the evening's menu interleaved in a sequence that, if followed to the letter, will produce the meal, with each dish appearing on the table at its appointed moment. Our chef, although gifted with an excellent set of taste buds, is extremely absentminded and can remember just one instruction at a time. Thus our hero plops a dozen potatoes on the counter, runs back to consult the list, peels the potatoes, dashes back to the list, cuts the potatoes into one-inch cubes, checks the list, and so on. A conventional—"serial"—computer, from the lowliest laptop to the mightiest mainframe, works in exactly the same way. The computer's central processing unit executes the recipe, or program, step by step—pulling data out of storage piece by piece, doing something to each one, and then putting it back before looking at the next instruction. The more powerful the computer, the faster the chef sprints. But that's not how the brain works at all.

The brain—any brain, from a slug's on up—is more like a medieval kitchen in a great lord's palace on the eve of a feast. A multitude of helpers bustles at a variety of tasks, shouting advice and instructions to one another. Everything happens all at once, and the banquet emerges almost spontaneously from the coordinated actions of many individuals. In the brain, these individuals are called neurons, and computer systems based on the notion of a network of simple devices acting collectively are called neural networks. The net's power lies in the interconnections, or synapses, between the neurons. The neuron itself is a threshold device that "fires"—generates an output—whenever its cumulative input exceeds its threshold. But one neuron in the human brain (and there are about 10^{11}—100 trillion—of them) may connect to 10,000 or more other neurons.

This redundant, highly interconnected scheme has other advantages. Returning to the kitchen for a moment, if some prankster crossed a line off of Super Chef's master list, the goose might get cooked unplucked, an error unlikely to happen in the castle kitchen. Or if Super Chef should fall down the wine-cellar stairs, the guests would go hungry that night. But if a few of the castle staff don't show up, no matter—the dinner still comes off. "Fault-tolerant algorithms" (an algorithm is a detailed strategy for attacking a problem) and "graceful performance degradation" as bits of hardware fail are hallmarks of neural nets but not, alas, of serial computers.

Furthermore, serial computers' ability to follow lots of step-by-step instructions very, very rapidly makes them dandy adding machines or tax auditors, but doesn't enable them to recognize Aunt Emma from a photograph, or, having recognized her, to remember that she and Uncle Joe have two children and a cabin in the mountains, and every other detail of their lives. It doesn't help a computer to reach for a pen to write her a note, either. These problems don't break down into cut-and-dried programs because there are just too many variations to list every contingency explicitly. But the connections in a neural network act like unexplicit rules. Patterns of associations—Aunt Emma, Uncle Joe, two kids, mountain cabin—become patterns of connections between neurons. The stronger a connection, the closer the association.

Given an input that matches some part of the pattern, the connections allow the net to retrieve the rest of it—a feat called associative memory. The connections feed back into one another, and signals slosh back and forth through the network along the pathways with the strongest connections. The feedback pulls out all the related information, regardless of which item you begin with—think of the mountains, and you'd still come up with Emma and Joe. This is called content-addressable memory. And the multiple connections can encode multiple memories, keeping Uncle Joe from being confused with your bachelor brother Joe. The connections can even cope with partially wrong input: if someone asked you about Aunt Emma's three children, you could set the record straight. Conventional computer memories, on the other hand, merely tuck away each tidbit of information in pigeonholes that bear no clues to the relationships between their contents.

People have known all along that computers and brains don't think alike, but biology's subtleties remain elusive.

Most things that neural nets do well—recognizing patterns such as faces, making decisions based on fuzzy or incomplete data, or displaying motor skills such as hand-eye coordination—require feedforward circuits as well as feedback. The theory is that information from an input layer of neurons trickles down through one or more layers of "hidden" neurons, again following the pathways of strongest connections, to an output layer. The hidden layers somehow filter the input, recognize the critical features needed to make a decision, and steer the system to the correct output, or stable state.

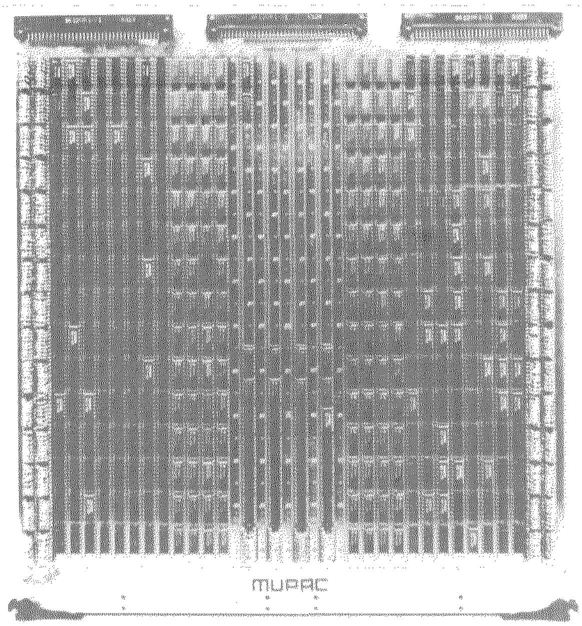
Neural nets, both man-made and biological, work fast. By exploring their options all at once, rather than scrutinizing each one in turn, they give you a pretty good answer immediately instead of the best possible answer in a week. And if the nets encounter a strange new input, they'll make an educated guess based on the information they have.

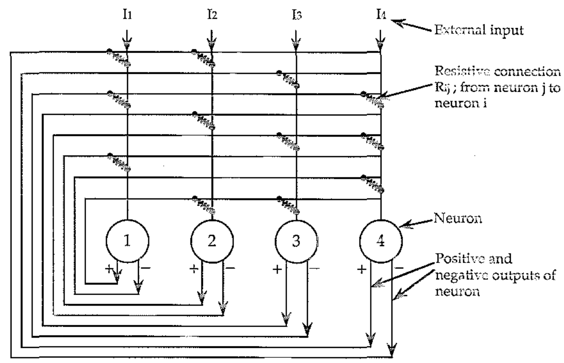
But the most remarkable feature of neural nets is that they can *learn* to do these things. A man-made neural network can alter its internal connections—strengthening some, weakening others—while being shown a training set of correct input-output pairs, until its outputs consistently match the right outputs. It may take a few thousand tries to get things right, depending on the problem's complexity, but still. . . .

If neural nets are so all-fired smart, why haven't they taken over? Why fool with serial computers at all? Because it's an awful lot easier for people to design and build something that does one thing at a time. People have known all along that computers and brains don't think alike, but biology's subtleties remain elusive. The first simple neural networks capable of learning were developed in the 1960s. Throughout the 1960s and 1970s, a handful of people labored, with some successes, to develop computational models of how real neurons behave, and to create the theoretical and mathematical underpinnings needed to design machines that could, in the broadest sense, mimic that behavior.

Neural network research began to catch fire in the 1980s for several reasons. Neurobiologists have made great strides in finding out how neurons work; computers have attained enough power to make running serial simulations of parallel processes practical, if time-consuming; and the development of analog VLSI (very-large-scale-integration) chips by Carver Mead (BS '56, MS '57, PhD '60), Caltech's Moore Professor of Computer Science, is enabling powerful synthetic neural nets (albeit puny ones compared to biology) to be graven in silicon. But the conceptual kindling was probably a 1982 paper by John Hopfield, the Dickinson Professor of Chemistry and Biology, showing that the same mathematical tools that physicists routinely use to analyze large physical systems with complex interactions, such as freezing liquids, could be applied to a network of binary (on/off) switches—a simple neural net. Hopfield demonstrated that a set of switches, each of which was wired to every other switch—fully interconnected feedback—acted as an associative memory; they're now called Hopfield memories. This revelation made neural nets accessible to the legion of scientists and engineers whose last contact with biology was probably a frog in a dissecting pan in high school, but who thought that neural nets might be applicable to their own computational problems, if they only knew how to handle them.

Even now, however, neural-net research lingers in the "not yet" stage, as in, "Can you make a neural net that can read a book and summarize its plot?" "Not yet." "Well, all right; literacy *is* a lot to ask. Can you make a mosquito brain that can dodge a midair swat?" "Not yet." In fact, the most successful multilayer neural-net application being sold commercially to date is a serial-computer program to evaluate loan applicants' creditworthiness. The program picked probable defaulters on small





Right: The DARPA-funded 32-neuron network.
Far right: A schematic drawing of fully interconnected feedback.

loans as accurately or better than did typical human loan officers. The number of mosquitos employed by S&Ls is unknown.

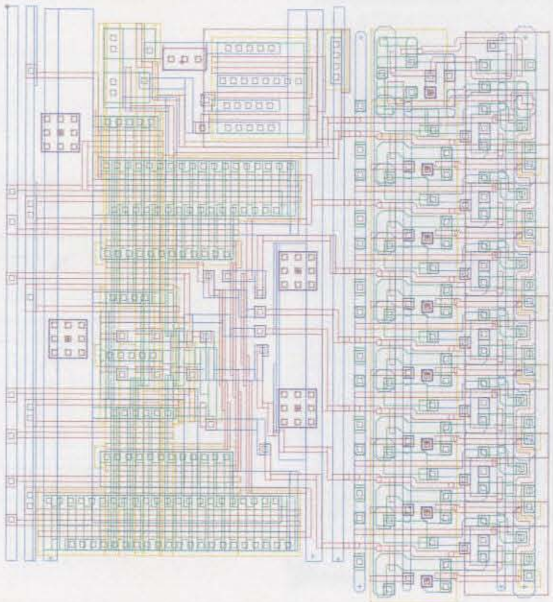
Caltech and JPL's history of neural-net collaborations goes back to 1981. John Lambe (now retired, but still returning bimonthly to JPL as a Distinguished Visiting Scientist) was visiting campus that spring as a Fairchild Distinguished Scholar on leave from the Ford Motor Company. He attended Hopfield's first Caltech seminar on neural nets, given in the improbable guise of an applied physics talk, and was smitten by their possibilities. Lambe left Ford for JPL that fall, and immediately built the world's first nonbiological feedback neural net—six neurons, interconnected with toggle switches, arrayed on an 18-inch-square Formica board. "The joke used to be that it stored two bits of information per square foot," recalls Anil Thakoor, now the head of JPL's Neuroprocessing and Analog Computing Devices Group. This network, far too small to do anything resembling computation, was nonetheless very useful. Lambe and Hopfield played with the toggle switches and found that each neuron assumed a predictable voltage—the network had settled into a stable state, in other words—instantaneously, allaying fears that any fully parallel man-made network would never settle down, but, a victim of the less-than-precise nature of its analog components and the stray capacitances in its hardware, would oscillate forever instead. This first model was built with JPL Director Lew Allen's blessing in its most tangible form—money from the director's discretionary fund.

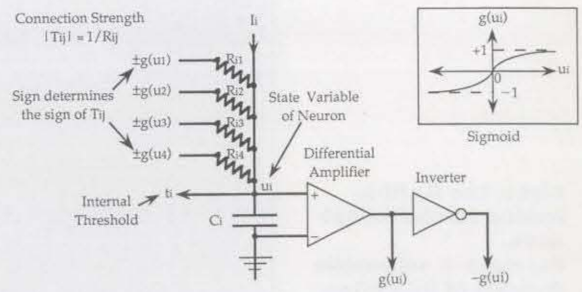
This network was big enough to address practical engineering concerns like power dissipation.

Lambe, Thakoor, and Alex Moopenn built a better net the following year, using Defense Advanced Research Projects Agency (DARPA) funds this time. This network was also 18 inches square and used off-the-shelf parts, but it had 32 fully interconnected neurons—1024 synapses. And instead of using toggle switches to make binary connections, this net used resistors to mimic the brain's variable-strength synapses. This network was big enough to address practical engineering concerns like power dissipation. Since the network's product is a distribution of voltages across its output neurons, any internal power loss affects it. Even when the connection between two neurons is supposed to be strong, you don't want a flood of current going through it, burning up power and converting it to heat—a good way to melt components, especially with lots of them active simultaneously. The group found that less was more—they could make good, strong connections out of a network of megohm (million-ohm) resistors, put one volt into the network, and get milliwatt power dissipation, well within the chip's comfort zone. And the output pattern was robust—if the initial voltage or some of the resistors were off by a few percent, the network came to the same stable output state as fast as ever.

Having demonstrated that building neural nets was actually practical, the next step was to try it in VLSI. As it happened, Mead and Hopfield were teaching a joint course, The Physics of Computation, that year—1983. (Mead, Hopfield, and the late Richard Feynman had initiated the course in 1981.) Two of Mead's grad stu-

A neuron is conceptually a very simple device, as the schematic to the far right shows—outputting "+1" if its input exceeds its positive threshold, "−1" if its input exceeds its negative threshold, and "0" if neither of the above is true. The reality is a trifle more complex, as shown by this integrated-circuit diagram of a seven-bit synapse, right.





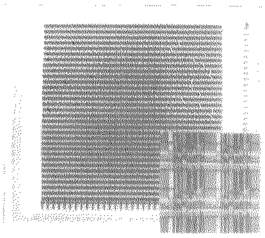
dents, Michael Emerling and Massimo Sivilotti (MS '86), designed a 22-neuron Hopfield memory chip, fully interconnected with adjustable-strength synapses, as a class exercise. A batch of the chips was actually built that December—the world's first single-chip VLSI neural nets. The chip could store three memories and retrieve any one of them from partial input in less than 50 microseconds (millionths of a second). One of the batch was severely defective—40 percent of its connections proved inoperative—yet it still held two memories, and retrieved them as quickly as its more able brethren. A larger chip followed the next year, and silicon neural nets were off and running.

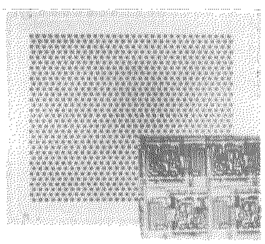
(Mead's VLSI has only partially solved the connectivity problem, however. If each neuron connects to every other neuron, then the number of synapses increases as the square of the number of neurons. A 64-neuron chip—large enough to actually do something useful—needs 4096 interconnections. (By comparison, a slug brain has between 100,000 and 1 million neurons.) The brain has the luxury of making connections in three dimensions, but a microchip is still essentially flat. Fortunately, less-than-full interconnection suffices for many simple applications. It also turns out to be wretchedly difficult to make a continuously variable resistor out of silicon. Digital synapses that use stairways of preset resistances are much easier, although their design is quite elaborate. An 8-bit synapse affords 256 steps, sufficient for most applications.)

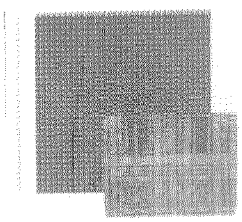
A critical mass of people had formed by 1983. The group included Hopfield; Lambe; Mead; Allen; Thakoor; Terry Cole (PhD '58), then a senior member of the technical staff at JPL and a senior research associate on campus and now JPL's chief technologist and a senior faculty associate on campus ("He knew an awful lot of people in both camps and was a very important part of the glue holding the critical mass together," says Hopfield); Demetri Psaltis, who joined the Caltech electrical engineering faculty in 1980; John Pierce (BS '33, MS '34, PhD '36), chief technologist at JPL since 1977 and professor emeritus of electrical engineering since 1980; Edward Posner, a visiting professor of electrical engineering and JPL's chief telecommunications and data acquisition technologist; and Robert McEliece (BS '64, PhD '67), professor of electrical engineering. The group began meeting regularly, if informally, to talk about neural nets, and hosted more formal sessions each October from 1983 through 1985, which drew biologists, engineers, and physicists from Lab and campus. Anyone doing neural-net work was encouraged to speak about it, and bound copies of the proceedings circulated widely up the mountain at JPL and down on campus.

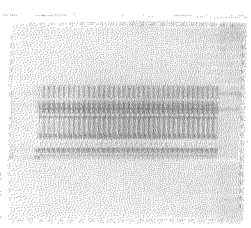
By 1986, JPL's neural hardware group, then under Satish Khanna, was well established, and had launched a program to build modular chips that could be wired together to make feedback, feedforward, or hybrid arrays of any size for specific applications. The group now has a family of neuron-only and synapse-only chips in the square, 84-pin format standardized by integrated-circuit manufacturers. (These chips, and every previous one all the way back to Sivilotti's

Some of JPL's family of neural chips. From the top: analog, binary, and seven-bit versions of a 32 × 32 synapse chip, and a 36-neuron chip.









A slug brain has between 100,000 and 1 million neurons.

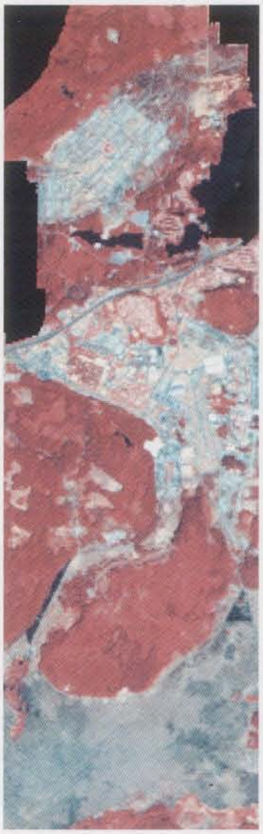
and Emerling's, are actually built by MOSIS, a government-funded custom-chip broker for defense-related research.) The hardware collaboration continues very closely today, with Mead a frequent visitor up the mountain, often in his capacity as a member of the Center for Space Microelectronics' scientific advisory board. "Even a quick question to Carver often gets us a significantly better solution to a problem in hand," says Thakoor. "He'll point out completely new directions to explore."

Also in 1986, Caltech established its Computation and Neural Systems PhD program, the first and, according to Posner, the most truly cross-disciplinary one of its kind in the world. The program drew faculty from biology, chemistry, engineering and applied science, physics, and mathematics. "It's easier to set these things up here than most places, because of Caltech's small size and fewer layers of bureaucracy," says Posner, "and that's one of our greatest strengths."

That same year, Cole and Allen established a neural-net theory group at JPL. The Neural Computation and Nonlinear Science Group, as it is properly called, is headed by Jacob Barhen, who is also a visiting associate in engineering and applied science on campus. "We have our own critical mass of very good people," says Barhen. "I think that we and Bell Labs are the only two places in the world that have enough nonlinear theorists working together to really make a difference. Some of the papers we have published have really revolutionized the field." The group does basic research in neural-network theory and develops algorithms for specific problems, driving the development of hardware to run them. In general, algorithm design begins with simulations running on an ordinary computer, and many algorithms, still too complex to build into hardware, remain there. The simulation calculates each neuron's output to its mates until the network reaches a stable state. Barhen's group is considerably better off than most, because their simulations run on the Cray X-MP supercomputer recently acquired as a JPL-Caltech joint facility.

The original critical mass lost some of its cohesiveness as more people got involved. Campus folk gravitated to the CNS program, and JPL folk, suddenly blessed with Defense Department money, became involved in their own formal programs. Some collaborations continued, but most people went their separate ways, exploring the terrain. Projects became complementary rather than collaborative. In the last year or so, however, a new generation of collaborations has sprung up to capitalize on the past few years' work. While most have been successful, a few have foundered because, in the words of one campus observer, "You need someone on the JPL end who is absolutely committed to the collaboration for it to work. There's a lot of bureaucratic overhead, mainly because JPL's research is contract-funded, while Caltech grants are usually unrestricted. When you're on contract funding, you have to break everything down into tiny increments, and you spend all your time writing progress reports." With that caveat duly noted, here are some of the successes.

JPL's most ambitious hardware project to



"There's a lot of ad-hockery involved now when you set up a network. We'd like to see how we could make a network grow itself into the appropriate structure as the data arrive."

date could be called an electronic scout. Given a false-color Landsat image of a piece of ground, the system finds the best way to drive a vehicle from one point in the image to another. It's essentially a data-compression problem. Each point, or pixel, of the image consists of 24 bits of information—eight each of red, green, and blue—but we just want one bit of information: can we drive through it or not? Three layers of neurons decide what kind of terrain each of the almost 17 million possible shades of color represents, using algorithms developed by Nevin Bryant, Niles Ritter, and Thomas Logan of JPL's Cartographic Applications Group. The net ultimately determines three components: slope, vegetation type, and load-bearing capacity. These three components reduce to one—the movement cost to pass through that pixel. This output passes to a quasi-neural chip, under development by JPL hardware handyman Silvio Eberhardt with Douglas Kerns, a graduate student of Hopfield's. The chip finds the path from point A to point B that incurs the lowest cumulative movement cost. A serial simulation of the chip works fine. The prototype chip itself, which will process 25 pixels at a crack, is scheduled for fabrication this summer.

On a much more modest scale, Eberhardt, Fernando Pineda (from the Lab's theory group), and Mead-Hopfield grad student Ronald Benson have designed a 4-neuron prototype chip that incorporates the learning algorithm, which normally runs on a digital computer, directly onto the chip. This "recurrent back-propagation" learning algorithm (there are other types) com-

Opposite page: A false-color Landsat Thematic Mapper image (left), combining three visible and one infrared spectral bands, of a portion of Ft. Lewis, Washington. The computer interprets the color, classifying the terrain as either forested (dark green), grassy (light green), urban (white), water (blue), or unclassifiable (black). A conventional program (middle) misread numerous land areas as water or urban, and couldn't figure out several large regions at all. The simulated neural network (right), although given only one-fourth as many correctly classified regions as exemplars, performed considerably better. Both programs had trouble with Nisqually Lake—the seahorse-shaped region in the lower left of each image—whose shallow, luxuriantly vegetated waters gave an ambiguous spectral signature.

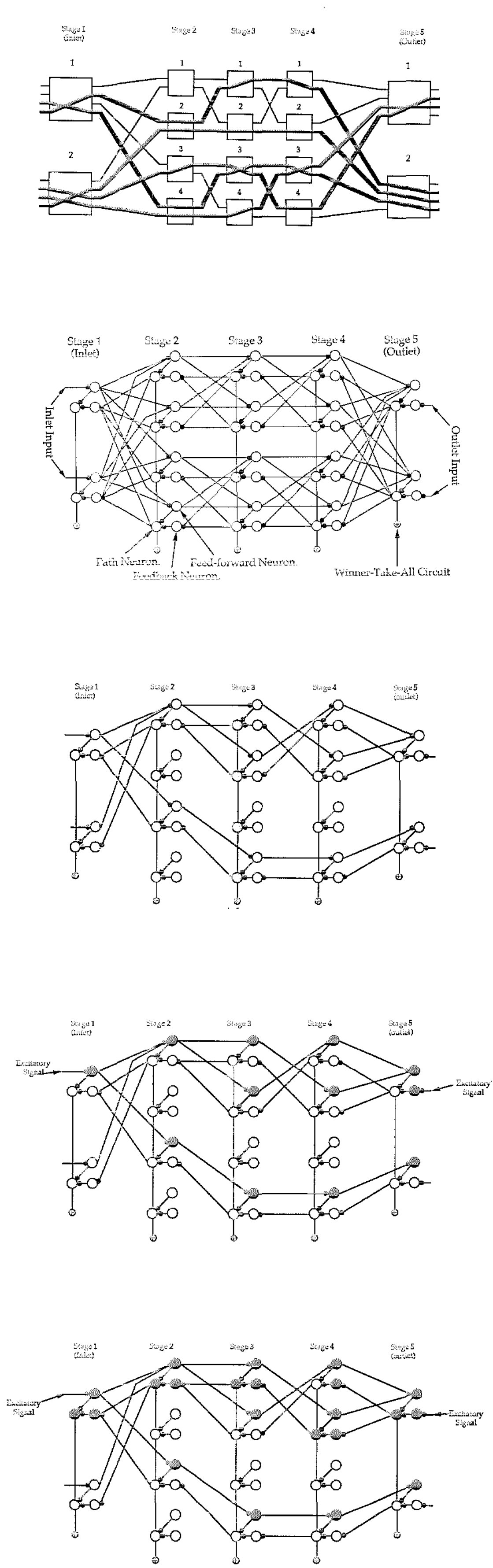
pares the network's actual output with the correct output and, with the input still in place, twiddles the connection weights from the output layer on back upstream until the outputs match. These algorithms usually clank through many cycles of software steps, but Pineda saw a way to restate the algorithm as a set of differential equations that could be transcribed directly into silicon. The neurons connect via "floating gate" transistors. The current through a transistor is governed by the amount of charge in the transistor's gate, which would have to be replenished if the gate were connected directly to the rest of the circuit. But a floating gate sits in splendid isolation. Charge injected into it by quantum-mechanical tunneling stays there for months.

The collaboration had a nice balance of forces. Floating gates—which are routine in some chip designs but had not been used extensively in neural nets—were Benson's specialty. He and Eberhardt came up with a circuit design based on Pineda's algorithm. The work began in November, 1989, and the chip was sent out for fabrication in the spring of 1990. Testing will begin in the fall. "Even if this version doesn't learn successfully, it will show us a lot about how the physics in the chip constrains the algorithms that can be put on it. It betters our odds of success with the next one," Pineda says.

Looking farther ahead, Padhraic Smyth (MS '85, PhD '88), of JPL's Communications Systems Research Group, would like to develop faster learning algorithms. Smyth did his graduate work with Associate Professor of Electrical Engineering Rodney Goodman and has kept in close touch with him since moving up the mountain. The two are now coprincipal investigators on a just-launched two-year project that will use techniques from information theory, probability theory, and statistics to try to discover exactly how neural networks learn. A network learns through trial and error, regardless of what algorithm adjusts the connections. But learning, like life, is a risky business. The learning algorithm somehow has to decide what factors in the input are critical to making the right decision. If the algorithm oversimplifies things, the network's "mental image" may not apply to all circumstances, but if the algorithm retains too much complexity and enshrines a host of extraneous factors in the net, it may not function at all. "We'd like to be able to determine the right network architecture just by looking at the data it's going to handle," says Smyth. "There's a lot of ad-hockery involved now when you set up a network. We'd like to see how we could make a network grow itself into the appropriate structure as the data arrive."

Eberhardt and colleagues Taher Daud and Raoul Tawel are working with Caltech Senior Research Associate in Theoretical Physics Tom Gottschalk and former Professor of Theoretical Physics Geoffrey Fox (now at Syracuse University) on a chip to solve "dynamic assignment" problems. The specific problem is this: if there are 25 missiles coming at you, and you have 25 missiles of your own to shoot back, how do you ensure that each good missile shoots down a different bad guy, and doesn't chase all over the sky in the process? You don't need to find the absolute best solution—the shortest path for each missile—in this situation, but you *really* need an answer fast. The neural net should settle into a reasonably good answer in a few millionths of a second. The chip, now in fabrication, will match 64 object pairs—skimming through a space of 2.2×10^{89} possible combinations to do so. Although the chip was developed for the Strategic Defense Initiative program, the problem is a generic one, appearing in such civilian guises as routing calls through a telephone exchange, or ensuring that all the processing units in a parallel computer are sharing the work equably.

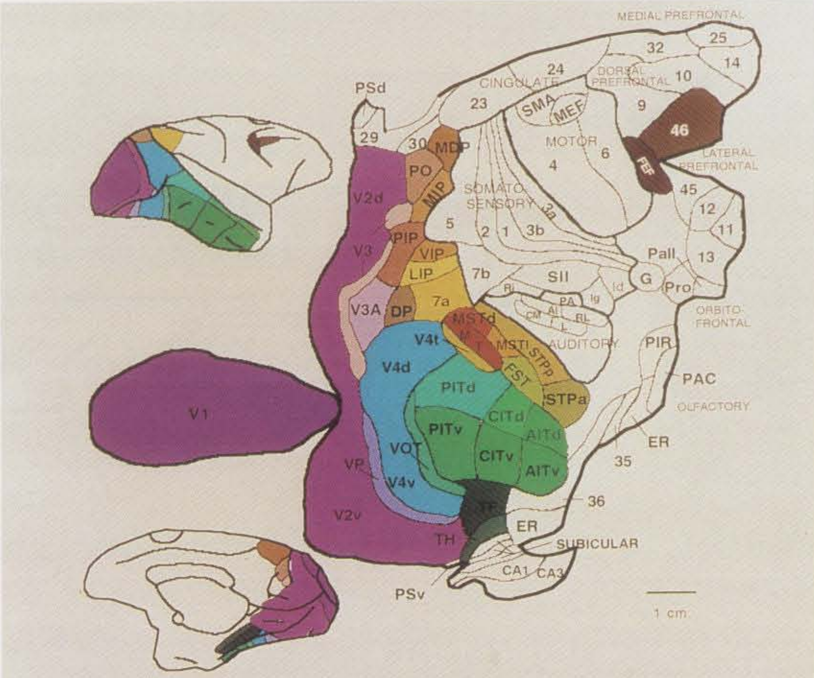
Posner, the communications technologist, is applying neural networks directly to communications problems by designing special-purpose nets whose architecture mimics the problem's structure. One of his grad students, Timothy Brown (MS '87, PhD '90), showed in his thesis that a certain neural circuit with inhibitory feedbacks does, in fact, solve the telephone routing problem quite nicely. All the members of one set of



Routing calls through a five-stage telephone exchange. Top: the relays are routing five calls in progress (heavy lines) between inlet-outlet pairs (1,2), (1,2), (2,1), (2,1), and (2,2). Next: A neural-net model of the same relay arrangement. Each neuron is shown as a large open circle with one or more output lines leading from it. A small filled circle is an output connection to an adjacent neuron. "Path Neurons" trace call routes. "Feedforward Neurons" are stimulated by the inlet stage, and in turn stimulate inactive path neurons that could carry the call forward through the network. "Feedback Neurons" are stimulated by the outlet stage, and stimulate inactive path neurons along a route leading back toward the inlet. A path neuron can only become active when stimulated by both a feedforward and a feedback neuron, thus tracing a continuous route through the exchange. "Winner-Take-All Neurons" inhibit competing path neurons, preventing each call from using more than one route. Center: The connections available to route additional calls; these connections correspond to the light lines in the top figure. Next: A call request for (1,1) turns on the feedforward neurons leading from Inlet 1. Active neurons are shown as filled circles. Bottom: The feedback neurons from Outlet 1 turn on, lighting up two available routes for the call. The winner-take-all neuron at Stage 4 arbitrarily chooses to route the call through Relay 2, lighting up one set of path neurons the rest of the way.

neurons are connected to each other in the same way that the exchange's relays are, and set up the call's route. A set of feedback neurons ensures that the routes don't interfere with each other. And a set of "winner-take-all" neurons ensures that each call only gets one route. "Most people are looking to neural nets to solve 'fuzzy' problems, like pattern recognition, where the things in the problem that are critical to solving it aren't well understood," says Brown. "This problem is very well understood. The phone company's computers have been solving it for years. But we can solve it much faster by building the computations right into the hardware. The network doesn't have to learn anything." Brown collaborated with Eberhardt, Daud, and Thakoor over the summer, trying to imbue his algorithm into an assemblage of the hardware group's standard chips. Brown found, as others have before him, that getting an algorithm to "take" in the hardware isn't as easy as it ought to be, but all is not lost—the JPL group was sufficiently impressed to hire him immediately upon his graduation.

And speaking of long-distance calls, there's the Communications Systems Research Section, in charge of developing the coding concepts and hardware and software prototypes that JPL needs to keep in touch with the Voyagers and other far-flung spacecraft. Neural networks, so adept at learning to recognize patterns, could prove useful for the error-correction and data-compression needed to send the data back to Earth.



The right hemisphere of a macaque brain, with the visual areas mapped in color, as seen from the right ear (small top figure) and from the left hemisphere (small bottom figure). The main figure shows the cortex unfolded and laid out flat. "V1" is the "primary visual area."

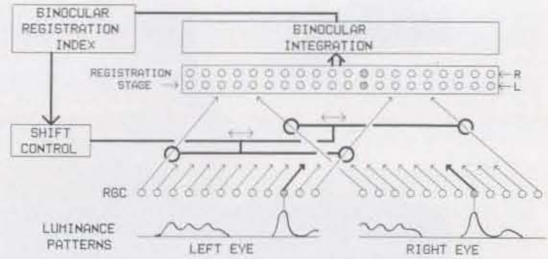
Kar-Ming Cheung (MS '85, PhD '87) and Fabrizio Pollara are working with Goodman on a neural-net data-compressor. JPL engineers used considerable ingenuity to cram all the Voyager data into the narrow communication channel available to it. But Voyager's data stream will be to the torrent of data from the next generation of spacecraft as a dripping faucet is to a fire hose. A single instrument on one of the two Earth Observing System (EOS) craft that will be watching our own planet for signs of global change (see "Observing Earth From Space," *E&S*, Winter '89) will be spewing 300 million bits of information Earthward every second. Voyager's tiny brain could compress data two- or threefold through such stratagems as not transmitting how bright a given pixel in an image was, but rather the difference in brightness from the previous pixel. A neural net might achieve 10 times more compression by handling pixels in rectangular blocks. The network would compare the block to a "code book" of standardized pixel blocks, like matching a wallpaper swatch to a pattern book, and would transmit the index number corresponding to the block that most closely matched the original. A serial computer would take an inordinate length of time to thumb through a code book big enough to guarantee that any input could be matched with minimal distortion, but the answer would tumble right out of a feedforward net whose connection strengths modeled the code book. This project, funded by the same director's discretionary fund that underwrote the Lab's first neural net, got under way last year.

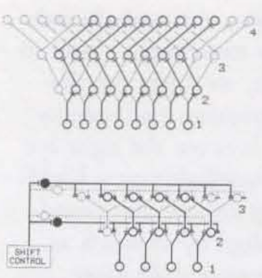
Meanwhile, down on campus, Professor of Biology David Van Essen (BS '67) is interested in neural nets for what they can tell him about real brains. Van Essen began as a traditional neurobiologist interested in primate vision. In 1985, he met physicist Charles Anderson (BS '57), then at RCA. "Charlie was looking at the same problems we were, but from the point of view of a device designer, and this gave him some novel ideas about how the visual system might work. When he moved to JPL in 1987, our collaboration increased in scope." Anderson, a senior member of the technical staff at JPL, is also a visiting associate in biology on campus. He and Van Essen are trying to discover why the world we see doesn't jump and wobble like a movie about to slip off the sprockets. It should—our eyes never sit still. Even when we stare fixedly at something, our eyeballs jink around in tiny involuntary movements. And in binocular depth perception, we judge the distance to an object whose position may vary by only a few seconds of arc between the right-eye and left-eye views. (The headlights on a Cadillac parked 200 miles away are one arc-second apart.) Yet our eyes misalign by as much as one-fifth of a degree, even stone-cold sober. How does the brain remove the gross errors and preserve the subtle differences?

The pathway from the eye to the visual cortex is generally thought of as hard-wired, with signals passing linearly along parallel columns of neurons in order to preserve relative-position information. But any individual retinal cell flickers on and off as the image dances across it, so if each retinal cell had a direct line to a particular cortical neuron, then the cortical "image" would be correspondingly unstable.

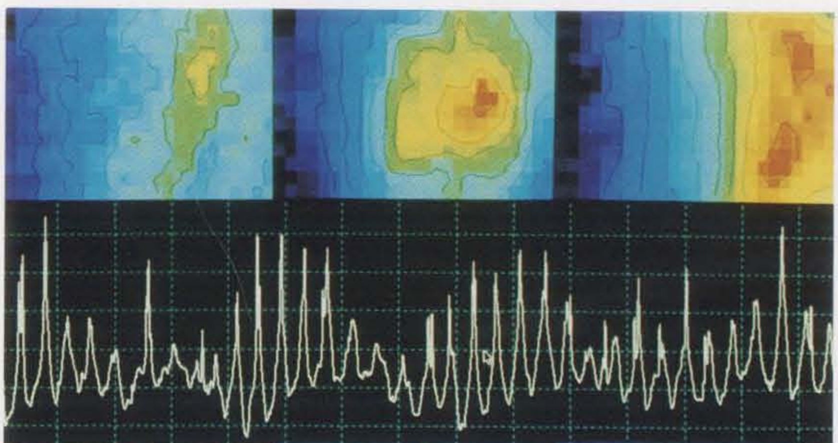
Anderson and Van Essen propose a pathway in which signals shift among columns to compensate for a wandering eye. In their neural-net simulation, the columns are sliced and layered like the pepperonis in a stack of frozen pizzas. Each pepperoni sends its output to a set of pepperonis in the pizza above, but not to the one pepperoni directly overhead. The connections go farther afield with each pizza. A set of inhibitory connections within each pizza suppresses shifts in all directions but one, keeping the parts of the image aligned. By tracing the right path through the pepperonis, the image can be kept in a fixed position in the cortex regardless of which retinal cells sent the signal.

The "shifter circuit" hypothesis is still very much in debate, but it does make testable pre-



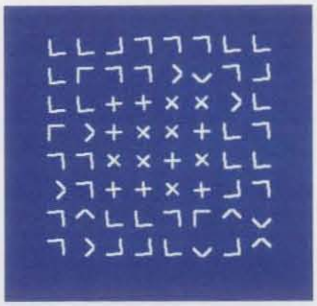


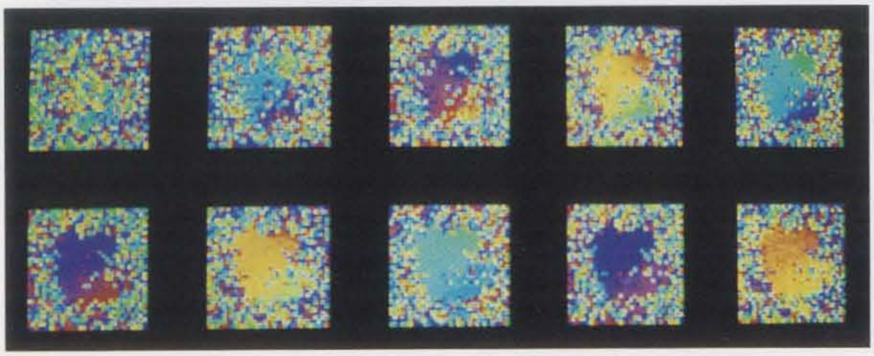
Right: Schematic showing how a shifter circuit could bring misaligned images from the eyes into proper registration in the cortex. The luminance peak from each eye is shifted until both peaks stimulate the same set of cells (hatched). Left: A simple shifter circuit. At every level, each cell stimulates two cells lying in opposite directions in the level above. The shift control suppresses activity along all but one set of paths (heavy lines) to align the final output correctly.
Below: A simulated olfactory-cortex oscillation pattern (left to right, top to bottom). Red regions are most active; blue, least. The central trace shows the simulated output from a single neuron.



dictions. For example, the image-shifting should occur as early as possible in the visual pathway, and certainly before the images from both eyes are fused for depth perception. Van Essen and grad students James Fox and Tobias Delbrück are studying the "primary visual area," where the first stage of visual processing in the cerebral cortex occurs, using microelectrodes that can localize the source of a nerve impulse to within one-tenth of a millimeter. The predicted shifts should be up to two or three millimeters, and thus readily detectable. Van Essen hopes to have a preliminary result within a year. "We're seeing something interesting going on. It's not exactly what the original theory suggested, but the visual cortex is definitely a more dynamic system than people have heretofore appreciated," he says.

Assistant Professor of Biology James Bower is also "reverse engineering" the nervous system—trying to discover how the brain's complex anatomy actually contributes to its complicated and subtle computations. His group is exploring the olfactory system, which has been mapped in considerable detail and contains elaborate hierarchies of dozens of cell types. The group is particularly interested in the contrasts between the olfactory system and the more elaborate and much more extensively studied visual system, which seems to work quite differently. The visual system reconstructs the three-dimensional world from two two-dimensional images, one on each retina. Every retinal cell responds, sending impulses to the primary visual cortex, where specific cells apparently recognize various attributes. Some cells fire, for example, when they



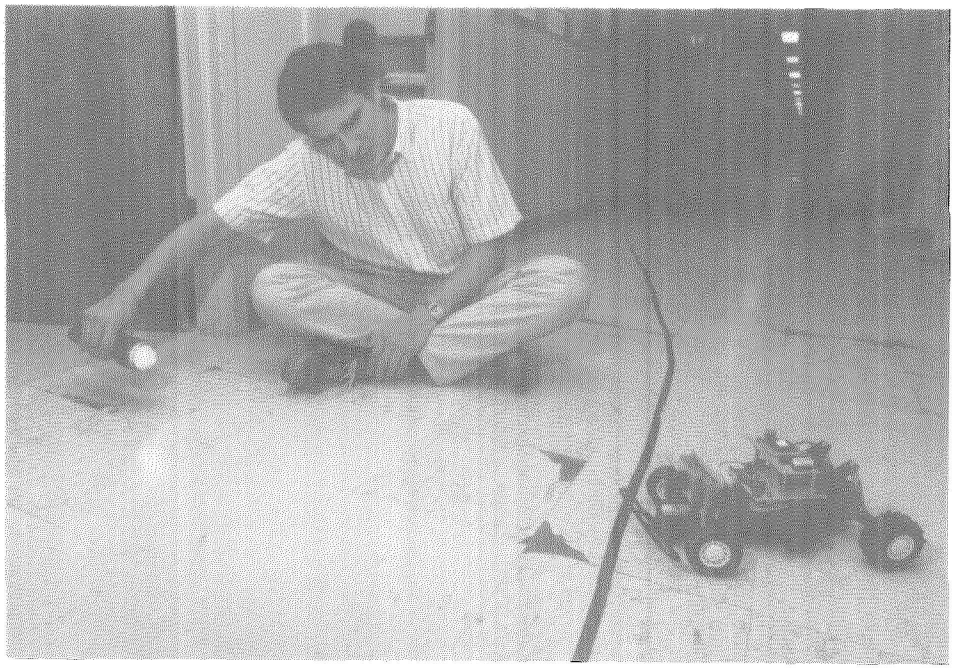


Right: Coupled random oscillators (below) begin to synchronize their firing patterns (colors) when the textural filters that drive them are shown a pattern (above).
Below: Baldi (holding mouse) and Bhalla use the enclosure in the background for their odor experiments.

perceive a vertical line, while others are triggered when an object moves from left to right. These attributes are hypothesized to get combined into objects in some complex manner farther on. But an odor has more than three dimensions, says Bower. "If you smell an apple pie fresh from the oven, your nose is sampling a set of volatile chemicals that will be substantially different from the set you'll sample if you smell the same pie after it has been sitting in the fridge for two weeks. But it still smells like apple pie." The membrane lining your nose—the epithelium—contains olfactory receptor cells that recognize and respond uniquely to millions of different volatile chemicals. The impulses travel to the olfactory bulb, where, instead of a particular neuron responding to "lemon" or "pine," many cells respond in some degree to many different inputs.

Pierre Baldi (PhD '86) of the Lab's theory group doubles as a visiting associate in biology, and is working with Bower on the mathematical theory behind a neural-net classification of odors. But Baldi, a theorist with degrees in psychology and mathematics, believes in getting his hands wet, too. "Biological phenomena are too complex for an experimentalist to be able to communicate everything to a theorist," he says. "If you try to be a pure theorist in biology, you'll miss the important details." Thus he, grad student Upinder Bhalla, and Assistant Professor of Biology Kai Zinn have started a set of behavioral experiments with mice. "Rats bite; mice don't," says Bhalla. "That's why we chose them." The trio are looking for a link between the olfactory system and the immune system. (This isn't so farfetched. Both systems recognize and respond to a bewildering variety of foreign substances, so why shouldn't they use similar methods? After all, nature is conservative—a successful stratagem often reappears elsewhere.) The mice learn to push one of two levers, depending on which of two odors wafts into their cage. The experiments will include normal mice and mice with defective T-cell receptors, an immune-system component that recognizes and binds to foreign matter. If the hypothesis is correct, the immunodeficient mice shouldn't be able to recognize as many odors, and an analysis of what they can't smell may reveal how their olfactory neurons are connected.

Baldi's taking a look at vision, too. He and Ron Meir, a postdoc in Hopfield's group, have just published a paper describing how the cortex might use differences in texture to discriminate between an object and its background. Their simulated neural net, which Meir calls "semi-biologically possible," is based on a recent German discovery that groups of neurons in the visual cortex fire simultaneously in "coherent oscillations." These oscillations may be how the cortex defines objects—all the neurons responding to features that are part of a chair would flash at one rate, while the neurons encoding the cat asleep on the chair would flash at a different rate from, or out of phase with, the chair neurons—not unlike having a video game in your head. The model consists of a series of filters tuned to recognize textural elements—vertical bars at a fixed separation, for example—and



Seated in the hallway outside the CNS lab, Koch tries to entice a light-seeking dune buggy, built by Andrew Hsu (BS '89) as his senior project.

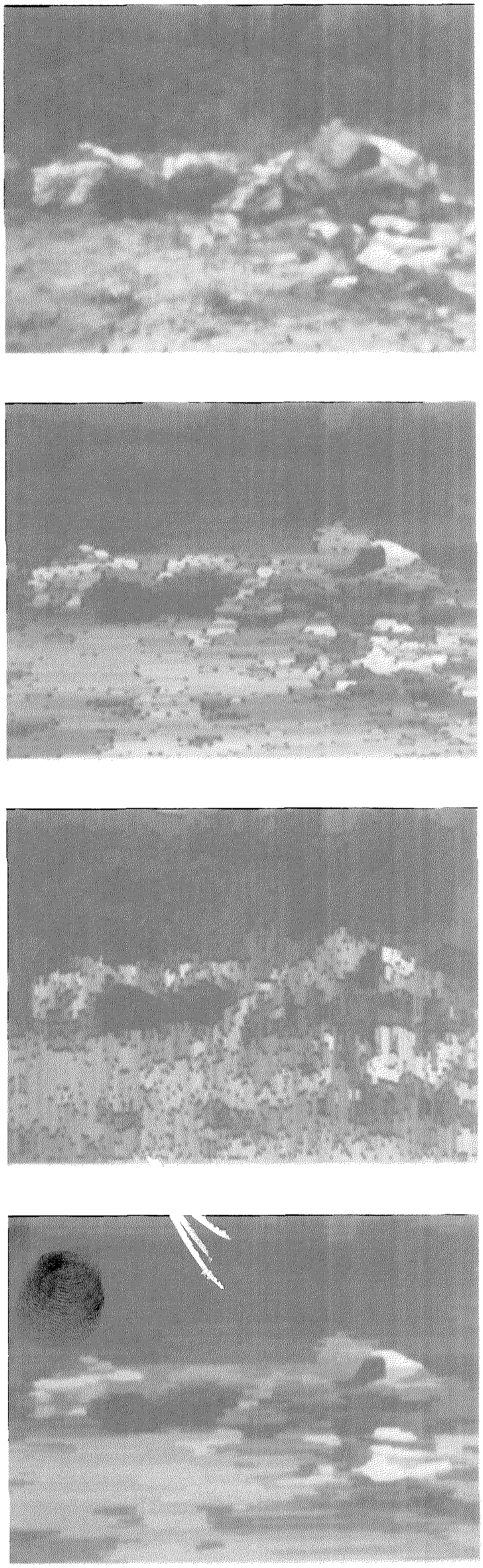
whose outputs drive arrays of coupled oscillators. When a serial-computer simulation of the neural-net model is shown a texture field—a pattern of plus signs on a background of Ls, for example—each oscillator begins to take note of its neighbors, and they spontaneously synchronize over a region corresponding to the pattern. The background remains random.

"These visual-cortex oscillations are very hot right now, because they've just been discovered, but we've known about them in olfaction for about 25 years, and there we think we know what part of the network causes them," says Bower. "We've constructed a biologically realistic simulation of that region with some 200 parameters to it. Pierre is using very abstract models, with three or four parameters, that are more tractable mathematically—exploring the problem unconstrained by biology. The two approaches feed into each other."

"We're essentially trying to put reflexes on chips."

Christof Koch, assistant professor of computation and neural systems, and his group have been working on another way of seeing things for the last four years (see "Computer's Eye View," *E&S*, Winter '88). The group designs "early vision" chips that do such basic jobs as deciding where an object's edges are, or, by calculating how fast those edges are expanding, when a rapidly approaching object will hit. (It's up to other, higher brain centers to identify the objects and figure out what to do about them.) When we look at something, even a boulder with a rough, textured surface, we see a uniform entity with distinct edges. These edges are discontinuities—the different colors of the boulder and the grass; the contrast between the sunlit rock and its shadow on the ground; or (Look out!) the downward motion of the boulder relative to the hillside. Each neuron in the chip corresponds to a pixel in a CCD (charge-coupled device) camera's visual field. The chip recognizes discontinuities—in color, light intensity, or relative motion, depending on the chip—and turns off all the neural connections that span the discontinuities, creating regions on the chip whose sizes and shapes correspond to the objects it "sees."

The group has built three chip generations based on this design, which derives from a retina chip designed by Mead grad student Michelle Mahowald (BS '85) in 1985, and has wired them into little vehicles that began life as radio-controlled toys. These seeing-eye dune buggies do their off-roading in the corridor outside Koch's lab. Although the chips are quite small—from 20 × 20 pixels up to 48 × 48 pixels, compared to some 360,000 in a home video camcorder—they can "see" well enough for the vehicles to zip along a line of black electrical tape on the white-tiled floor, or drive toward a flashlight in a darkened hallway. The group hopes one day to develop a system smart enough to maneuver a vehicle over a three-dimensional landscape. JPL's Brian Wilcox, a member of the vision-system design team for the proposed Mars Rover project, is designing algorithms that recognize and avoid obstacles. "We're essentially trying to put reflexes on chips," says Koch, "decisions that now have to be made by a central processing unit but that should really be made by a



A rocky slope as seen through Lawton's vision algorithms. The original scene is at the top, followed by the set of horizontal line segments, the set of vertical line segments, and then the object map at the bottom.

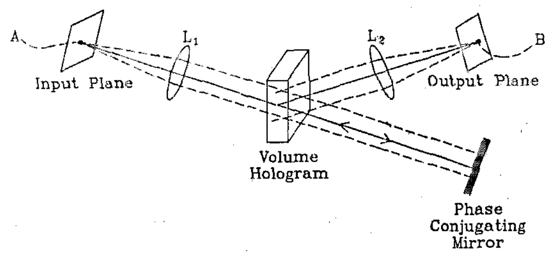
much lower level in the system."

If the Mars Rover never gets off the ground, there are still plenty of applications closer to home. Edge-detection chips could double-check that a bottling line is really putting two liters of soda in every bottle, or see that toilet paper winds evenly on the roll. And, realistically, "there's a lot bigger market for toilet paper than there is for Mars Rovers," says Koch.

Teri Lawton, of the Lab's theory group and the Mars Rover team, has yet a third perspective on vision. Lawton is using Caltech's CNS lab facilities to design and test "object-oriented" vision algorithms based on biological neural networks. Unlike other, pixel-based approaches, Lawton's algorithms divide a scene into regions with common properties—similar textures and gray-scale values, for example. The algorithms begin by compensating for the jouncing ride over uneven terrain, somewhat as we coordinate our eye and head movements to keep the eyes on one spot as the head moves. (JPL's robotics lab developed these pitch-, heading-, and roll-correction algorithms in the 1970s.) The scenes, now containing just those differences due to the vehicle's real horizontal motion, pass through two sets of filters. One set recognizes horizontal and vertical line segments. The other set registers gray-scale brightness. The gray scale automatically adjusts itself within shadows—which other algorithms perceive as flat, dark objects—to reveal smaller rocks that could wreck a rover. The algorithm then defines and remembers objects as two-dimensional closed loops made of overlapping line segments of roughly the same gray scale, penciling in obscured line segments where necessary. As the rover rolls along, it constantly updates its memory. Parallax—how much an object moves compared to the rest of the scene—gives each object depth, and allows the rover to build a three-dimensional picture, or "object map," of its surroundings. Because this picture includes several sensory dimensions—size, position, texture, illumination, gray scale, and motion parallax—it's more fault-tolerant than object maps depending on one parameter, such as edges, alone; and because the algorithm exploits the relative motion of whole objects in successive images, it handles diverse objects better and operates faster than pixel-based methods that must compute the change at every pixel between successive images. Lawton's simulations take about 12 seconds to generate a depth map, much faster than other algorithms, and the chip she eventually hopes to build with Mead's group should do the same job as quick as a wink. In the meantime, Lawton has been working—first with Brian Fox, a staff member in Koch's group, and now with Aaron Emigh, a senior from UC Santa Cruz on campus for the summer as part of Caltech's Summer Undergraduate Research Fellowship (SURF) program—to optimize the algorithm before committing it to silicon. "Caltech has made it possible for me to do this work," says Lawton. "I couldn't have done it otherwise. And it's a contribution to biology as well—we can use these three-dimensional terrain maps of natural scenes as a test bed to learn how the brain generates a three-dimensional world view from a two-dimensional retinal image."

Lawton's work can help the partially sighted on Earth as well, especially those people—mostly elderly—who can't see fine detail any more. Closed-circuit TV "readers" that magnify and brighten printed matter are already available. Readers modified with one of Lawton's algorithms automatically enhance the text to match the user's remaining contrast sensitivity, and render the text in shades of gray more easily perceivable than black and white. Users have experienced a two- to fourfold increase in reading speed at up to 70 percent less magnification than they needed to read text on the old machine. Caltech has optioned the patents to the gray scale portion of the system, which could be commercially available soon. The next logical step, electronic spectacles (lightweight, wearable units with CCD cameras to look at the world, some simple electronics to process the image, and liquid-crystal displays—LCDs, the screens used in tiny TVs—to present the result) is well within the reach of current technology.

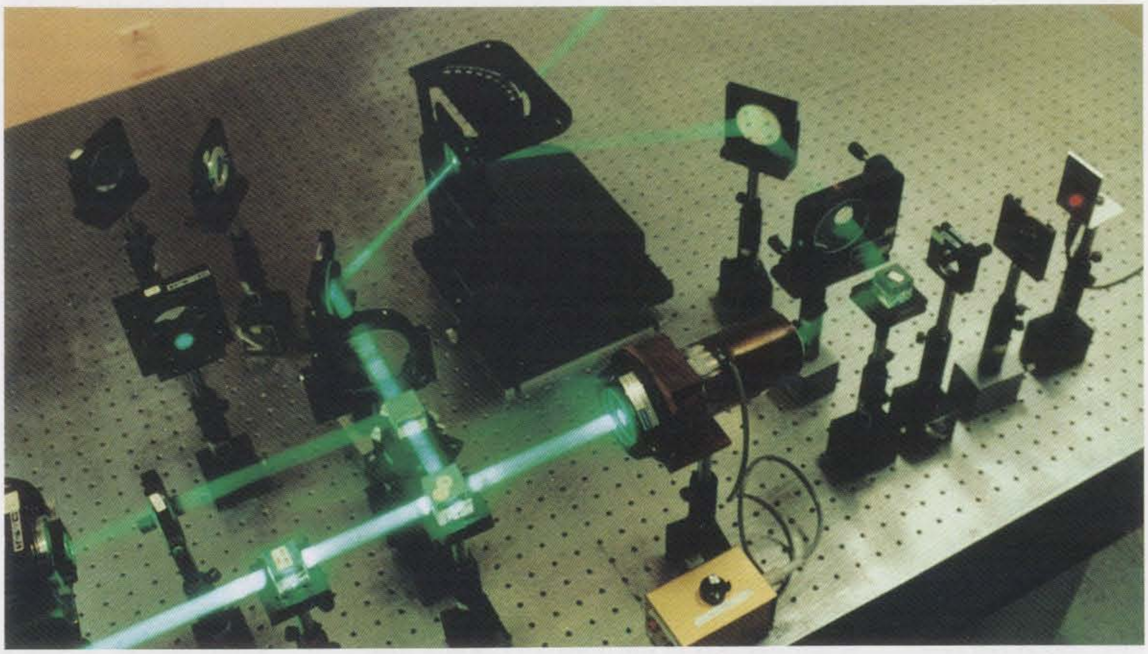
Hybrid neural nets using electronic logic and optical interconnects may be practical shortly.



General scheme of a holographic associative memory. L_1 and L_2 are lenses. The "Phase Conjugating Mirror" is used when updating the memory.

Although connections make the net, you can't pass wires through each other. Beams of light, though, can intersect and carry information without interfering with one another. Optics and electronics can mate, with LCDs or LEDs (light-emitting diodes) converting electrons into photons, while photovoltaic or photoconductive devices convert light back into electricity. The optical equivalents of logical processors are still rudimentary, so full-fledged optical computers are still a gleam in the eye, but hybrid neural nets using electronic logic and optical interconnects may be practical shortly. Professor of Electrical Engineering Demetri Psaltis, who maintains an office at JPL and consults there one day a week, thinks that "these hybrid systems will prove very useful in the next few years. Neural nets are ideally suited to coping with the real world, in robotic control, for instance. This way you can have a very highly interconnected net that interprets sensory inputs and passes its conclusions on to a serial controller that decides what to do. And you could have another net taking the controller's output to actually guide the robot."

Psaltis's group is working on hybrids. Grad student Steven Lin is collaborating with Jae Kim of JPL's Microdevices Lab to build neural arrays of gallium arsenide, a semiconductor faster than silicon that can also emit light. (Both semiconductors can be photodetectors.) Each neuron incorporates a light detector and a light emitter. The emitters shine up from the chip's surface into a hologram. The hologram can be two-dimensional—an optical disk, like a CD—or



Above: A holographic-memory loop. Input comes from the far right, where the red light illuminates a transparency, projecting its pattern into one end of a liquid-crystal light valve—the flashlight-shaped object at center. A laser beam from the lower left is reflected off the valve's other end according to the pattern. A cube-shaped beam splitter diverts the patterned beam back to a holographic medium in the angle-calibrated mounting at rear. The hologram's output emerges at an angle and goes back to the light valve's input side to complete the feedback loop. Thus the light valve acts as the set of neurons, using an external input and the product of its own interconnections to generate an output. The output registers on a CCD camera at lower left, behind the incoming laser beam. The rest of the setup is used for training the memory.
Left: Arabic and Chinese numeral input-output pairs stored holographically.

a three-dimensional photorefractive crystal. (Such a crystal's refractive index—the degree to which it bends light—is itself light-sensitive. A powerful beam of the right frequency alters the crystal's electronic structure, and thus its refractive index. The change persists after the beam is gone.) The chip-produced hologram channels light from each emitter to each detector in proportion to the connection strength between those two neurons.

Disks are easier to work with at the moment, the technology being more mature, but they aren't really reprogrammable yet. Grad student Alan Yamamura is using disks to make a single layer of neurons act like a multilayer network. The disk stores each layer's connection strengths sequentially and spins in sync with the information flow from layer to layer. JPLer Jeffrey Yu (BS '83, MS '84, PhD '88, and a former student of Psaltis's) is working with Psaltis to apply this technique to image recognition.

The crystals are fully reprogrammable and, being 3-D, can store information more compactly. A crystal can be loaded holographically, for example, so that shining an Arabic numeral onto one face causes the corresponding Chinese numeral to shine out from another face. Scientists elsewhere have recorded more than 1000 such associations on a crystal. In theory, a crystal can store as many as several thousand images per cubic centimeter, versus the tens of thousands of images that would cover a five-inch disk. Even when crystal technology matures, however, it may not displace disks altogether. Crystal memories can fade as new memories are stored because each new light beam irradiates the entire crystal, partially obliterating its predecessors' traces. But a tightly focused laser writes memories on a disk with plenty of elbow room between them.

Meanwhile, the connection problem may have been solved by Senior Research Fellow in Applied Physics Aharon Agranat and grad student Charles Neugebauer (BS '88) in the group led by Amnon Yariv, the Myers Professor of Electrical Engineering and professor of applied physics. They have a chip that uses a CCD to store connection weights—a radical departure from its designed use as a light sensor. A row and a column of neurons adjoin the CCD, each pixel of which contains a dollop of electrons proportional to the connection strength between the corresponding row neuron and column neuron. (See "Photographic Memory," *E&S*, Winter '88.) The current version has 256 neurons, each of which connects to the other 255, and a thousand-neuron chip is well within reach of standard CCD technology. Agranat and Neugebauer are now building a computer board that will carry the chip and that can be plugged into any IBM-PC-compatible computer. Real neural nets, instead of just software simulations, will become accessible to thousands of researchers.

"Our current models don't allow you to remove just one association without affecting all the others, but animals do it all the time. It's the only way to cope with a complex, constantly changing world."

Yariv's group began collaborating with Barhen's group this year to see how easily their hardware and algorithms integrate. Their first project will be an algorithm to calculate discrete Fourier and Hartley transforms—the two most important (and, coincidentally, most computationally intensive) mathematical tools used in signal processing. The system could be used to process seismic data or hunt for gravitational waves, and might also come in handy in JPL's image-processing work.

And then there's robotics. JPL has been doing robotics all along, of course—strictly speaking, any autonomous spacecraft is a robot—but the Lab is also working on more traditional robotics problems. Joel Burdick, assistant professor of mechanical engineering, is starting several collaborations between his graduate students and various robotics groups on Lab. One student, Bedri Cetin, is working with Barhen to apply neural nets to optimization problems such as making a robot arm move efficiently. Cetin developed a new approach to the problem, based on recent work by Barhen and fellow group members Nikzad Toomarian and Michail Zak, that Barhen calls "a major breakthrough in optimization theory. Everything eventually becomes an optimization problem, so the payoff will be tremendous."

Optimization problems can be thought of as rugged landscapes of hills and valleys. Whatever the physical aspects of the problem—moving a robot arm around obstacles, finding the shortest route through 11 cities—the problem can be cast as a mathematical landscape (in more than three dimensions, if need be), wherein the lowest point in the deepest depression is the optimum answer. Finding this nadir is the mathematical equivalent of setting a boulder loose and waiting for it to come to rest. Various strategies, such as dropping several boulders all across the landscape, have been developed to ensure that you really do find the very deepest point. The breakthrough incorporates ideas from quantum mechanics—the boulder can "tunnel" through a mathematical hillside to escape from an exitless valley—and from nonlinear dynamic systems theory, wherein a newly discovered entity called a "terminal repeller" can suddenly give the boulder a shove strong enough to send it skittering to anyplace in the landscape. "This method has solved some standard optimization problems 100 to 1,000 times faster than the best competing methods," says Barhen. "And applying the terminal repeller concept to man-made neural nets allows them to do things they couldn't do before, like selectively forgetting old associations, or spontaneously creating new ones without extensive training. Our current models don't let you remove just one association without affecting all the others, but animals do it all the time. It's the only way to cope with a complex, constantly changing world."

A lot of people are trying to help robots cope with the real world. Robots to date have been pretty simple-minded creatures. Today's state-of-the-art industrial robot—or spacecraft, for that matter—is really more like a complex machine tool. It has to have nearly every gesture spelled out for it explicitly, and must work in a simple environment in which a few known objects occupy predetermined locations and everything else stays out of the way. But future NASA robots, the ones that will go day-tripping across other worlds or work on the space station, will have to think for themselves and adapt to a complex, changeable environment.

Carl Ruoff, a longtime member of the Lab's Robotics and Automation Section and now a graduate student at Caltech as well, is working with Professor of Mechanical Engineering Fred Culick on a rudimentary robotic Little Leaguer that can learn basic motor skills on its own in a simplified version of such an environment. The device will acquire hand-eye coordination: it will learn to catch (or hit) any ball—from a golf ball

Hollywood epics notwithstanding, day labor at 380 miles up is difficult, dangerous, and time-consuming.

to a beach ball—no matter how the ball is thrown. The robot will have to integrate artificial-vision data about its surroundings with tactile-, force-, and body-sensor data about itself. It will learn coordination the way kids do—first learning how to control its arm, and how the arm should "feel" in particular situations, then learning how the ball behaves. Once the robot has watched enough tosses to be able to predict their trajectories, it learns to intercept them. The team expects to have a simulation running on the JPL–Caltech Cray come August. It's a long way from a ball-catching machine to RoboCub. But the Little Leaguer, when operational, will embody some of the basic attributes that autonomous space robots will need.

Burdick, Barhen, Sandeep Gulati (also of the theory group), and Robotics and Automation Section members Charles Weisbin, Subramanian Venkataraman, Guillermo Rodriguez, and Hamayoun Seraji (who is also a lecturer in mechanical engineering on campus), have started an informal collaboration to think about the rest of these attributes. "The whole task of integrating all these functions—motor skills, sensory processing, memory, and a host of other things—into a system that can learn on the job and make the internal changes it needs to complete its mission, is extremely ambitious," says Gulati. Adds Venkataraman, "It's learning to adapt a mastered skill to different environments, unlike today's robots that would have to start from scratch with every new situation."

The group has chosen to work on an astronaut's apprentice as their demonstration project. Astronauts will be spending a lot of time outdoors in the next century, working on the space station and making service calls on satellites. Hollywood epics notwithstanding, day labor at 380 miles up is difficult, dangerous, and time-consuming. A buddy can't just toss you a Phillips-head screwdriver, for one thing. So JPL envisions self-propelled, voice-controlled robot gofers to fetch tools, maneuver bulky parts and hold them in position, and rescue free-floating objects (including astronauts) before they drift away. A helper taking orders from a human in this situation actually faces an environment more complex than does a solitary explorer picking its way among Martian crevasses to take rock samples, because the helper has to be aware of many objects in three dimensions traveling in all directions at once, including unpredictable humans that will blunder into its way.

Such a robot will need all the neural-network attributes described in this article and then some. It will need pattern-recognition skills and a flexible memory to understand spoken commands issued by many voices, acute vision and deft limbs to execute those commands, and a sophisticated "brain" that can plan complex tasks in a free-form environment.

Many years will pass before such a system can be built, but the group is planning to take the first step. Over the next two years, they propose to develop a system that can deal with uncertainty in a limited environment. The device, initially two robot arms bolted to the floor, will grasp one end of some large, perhaps flexible, object. A person would hold the other end, and a tug-of-war would ensue. The human would push and pull on the object, shift grips, and sometimes let go altogether. The robot would try to keep its end level at all times, and would have to adjust its response constantly to compensate for the human's actions.

There's a long way to go before an autonomous, adaptive robot's gray matter can be trusted in space. "Real biological networks have much complex internal structure that we don't understand," says Ruoff. "Large, complicated systems are really qualitatively different," adds Culick. "Building lots of little pieces and having them all work separately is one thing, and putting an integrated system together and making it work is quite another. It is, however, something that JPL has learned to do very well." The neural net or hybrid neural-serial system that ultimately results—if one does—may finally be the mosquito's intellectual equal. Then it may fairly be said that the Caltech–JPL connection will have helped neural nets come of age.□—*DS*