BUILDING BRAINS FOR BODIES

Rodney Allen Brooks and Lynn Andrea Stein
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We describe a project to capitalize on newly available levels of computational resources in order to understand human cognition. We will build an integrated physical system including vision, sound input and output, and dextrous manipulation, all controlled by a continuously operating large scale parallel MIMD computer. The resulting system will learn to "think" by building on its bodily experiences to accomplish progressively more abstract tasks. Past experience suggests that in attempting to build such an integrated system we will have to fundamentally change the way artificial intelligence, cognitive science, linguistics, and philosophy think about the organization of intelligence. We expect to be able to better reconcile the theories that will be developed with current work in neuroscience.

Project Overview

We propose to build an integrated physical humanoid robot including active vision, sound input and output, dextrous manipulation, and the beginnings of language, all controlled by a continuously operating large scale parallel MIMD computer. This project will capitalize on newly available levels of computational resources in order to meet two goals: an engineering goal of building a prototype general purpose flexible and dextrous autonomous robot and a scientific goal of understanding human cognition. While there have been previous attempts at building kinematically humanoid robots, none have attempted the embodied construction of an autonomous intelligent robot; the requisite computational power simply has not previously been available.

The robot will be coupled into the physical world with high bandwidth sensing and fast servo-controlled actuators, allowing it to interact with the world on a human time scale. A shared time scale will open up new possibilities for how humans use robots as assistants, as well as allowing us to design the robot to learn new behaviors under human feedback such as human manual guidance and vocal approval. One of our engineering goals is to determine the architectural requirements sufficient for an enterprise of this type. Based on our earlier work on mobile robots, our expectation is that the constraints may be different than those that are often assumed for large scale parallel computers. If ratified, such a conclusion could have important impacts on the design of future sub-families of large machines.

Recent trends in artificial intelligence, cognitive science, neuroscience, psychology, linguistics, and sociology are converging on an anti-objectivist, body-based approach to abstract cognition. Where traditional approaches in these fields advocate an objectively specifiable reality—brain-in-a-box, independent of bodily constraints—these newer approaches insist that intelligence cannot be separated from the subjective experience of a body. The humanoid robot provides the necessary substrate for a serious exploration of the subjectivist—body-based—hypotheses.

There are numerous specific cognitive hypotheses that could be implemented in one or more of the humanoids that will be built during the five-year project. For example, we can vary the extent to which the robot is programmed with an attentional preference for some images or sounds, and the extent to which the robot is programmed to learn to selectively attend to environmental input as a by-product of goal attainment (e.g., successful manipulation of objects) or reward by humans. We can compare the behavioral result of constructing a humanoid around different hypotheses of cortical representation, such as *coincidence detection* versus *interpolating memory* versus *sequence seeking in counter streams* versus *time-locked multi-regional retroactivation*. In the later years of the project we can connect with theories of consciousness by demonstrating that humanoids designed to continuously act on immediate sensory data (as suggested by Dennett's *multiple drafts* model) show more human-like behavior than robots designed to construct an elaborate world model.

The act of building and programming behavior-based robots will force us to face not only issues of

interfaces between traditionally assumed modularities, but even the idea of modularity itself. By reaching across traditional boundaries and tying together many sensing and acting modalities, we will quickly illuminate shortcomings in the standard models, shedding light on formerly unrealized sociologically shared, but incorrect, assumptions.

Background: the power of enabling technology

An enabling technology—such as the brain that we will build—has the ability to revolutionize science. A recent example of the far-reaching effects of such technological advances is the field of mobile robotics. Just as the advent of cheap and accessible mobile robotics dramatically altered our conceptions of intelligence in the last decade, we believe that current high-performance computing technology makes the present an opportune time for the construction of a similarly significant integrated intelligent system.

Over the last eight years there has been a renewed interest in building experimental mobile robot systems that operate in unadorned and unmodified natural and unstructured environments. The enabling technology for this was the single chip micro-computer. This made it possible for relatively small groups to build serviceable robots largely with graduate student power, rather than the legion of engineers that had characterized earlier efforts along these lines in the late sixties. The accessibility of this technology inspired academic researchers to take seriously the idea of building systems that would work in the real world.

The act of building and programming behavior-based robots fundamentally changed our understanding of what is difficult and what is easy. The effects of this work on traditional artificial intelligence can be seen in innumerable areas. Planning research has undergone a major shift from static planning to deal with "reactive planning." The emphasis in computer vision has moved from recovery from single images or canned sequences of images to active—or animate—vision, where the observer is a participant in the world controlling the imaging process in order to simplify the processing requirements. Generally, the focus within AI has shifted from centralized systems to distributed systems. Further, the work on behavior-based mobile robots has also had a substantial effect on many other fields (e.g., on the design of planetary science missions, on silicon micro-machining, on artificial life, and on cognitive science). There has also been considerable interest from neuroscience circles, and we are just now starting to see some bi-directional feedback there.

The grand challenge that we wish to take up is to make the quantum leap from experimenting with mobile robot systems to an almost humanoid integrated head system with saccading foveated vision, facilities for sound processing and sound production, and a compliant, dextrous manipulator. The enabling technology is massively parallel computing; our brain will have large numbers of processors dedicated to particular sub-functions, and interconnected by a fixed topology network.

Scientific Questions

Building an android, an autonomous robot with humanoid form, has been a recurring theme in science fiction from the inception of the genre with Frankenstein, through the moral dilemmas infesting positronic brains, the human but not really human C3PO and the ever present desire for real humanness as exemplified by Commander Data. Their bodies have ranged from that of a recycled actual human body through various degrees of mechanical sophistication to ones that are indistinguishable (in the stories) from real ones. And perhaps the most human of all the imagined robots, HAL-9000, did not even have a body.

While various engineering enterprises have modeled their artifacts after humans to one degree or another (e.g., WABOT-II at Waseda University and the space station tele-robotic servicer of Martin-Marietta) no one has seriously tried to couple human like cognitive processes to these systems. There has been an implicit, and sometimes explicit, assumption, even from the days of Turing (see Turing (1970)*) that the ultimate goal of artificial intelligence research was to build an android. There have been many studies relating brain models to computers (Berkeley 1949), cybernetics (Ashby 1956), and artificial intelligence (Arbib 1964), and along the way there have always been semi-popular scientific books discussing the possibilities of actually building real 'live' androids (Caudill (1992) is perhaps the most recent).

This proposal concerns a plan to build a series of robots that are both humanoid in form, humanoid in function, and to some extent humanoid in computational organization. While one cannot deny the romance of such an enterprise we are realistic enough to know that we can but scratch the surface of just a few of the scientific and technological problems involved in building the ultimate humanoid given the time scale and scope of our proposal, and given the current state of our knowledge.

The reason that we should try to do this at all is that for the first time there is plausibly enough computation available. High performance parallel computation gives us a new tool that those before us have not had available and that our contemporaries have chosen not to use in such a grand attempt. Our previous experience in attempting to emulate much simpler organisms than humans suggests that in attempting to build such systems we will have to fundamentally change the way artificial intelligence, cognitive science, psychology, and linguistics think about the organiza-

*Different sources cite 1947 and 1948 as the time of writing, but it was not published until long after his death.

tion of intelligence. As a result, some new theories will have to be developed. We expect to be better able to reconcile the new theories with current work in neuroscience. The primary benefits from this work will be in the striving, rather than in the constructed artifact.

Brains

Our goal is to take advantage of the new availability of massively parallel computation in dedicated machines. We need parallelism because of the vast amounts of processing that must be done in order to make sense of a continuous and rich stream of perceptual data. We need parallelism to coordinate the many actuation systems that need to work in synchrony (e.g., the ocular system and the neck must move in a coordinated fashion at time to maintain image stability) and which need to be servoed at high rates. We need parallelism in order to have a continuously operating system that can be upgraded without having to recompile, reload, and restart all of the software that runs the stable lower level aspects of the humanoid. And finally we need parallelism for the cognitive aspects of the system as we are attempting to build a "brain" with more capability than can fit on any existing single processor.

But in real-time embedded systems there is yet another necessary reason for parallelism. It is the fact that there are many things to be attended to, happening in the world continuously, independent of the agent. From this comes the notion of an agent being situated in the world. Not only must the agent devote attention to perhaps hundreds of different sensors many times per second, but it must also devote attention "down stream" in the processing chain in many different places at many times per second as the processed sensor data flows through the system. The actual amounts of computation needed to be done by each of these individual processes is in fact quite small, so small that originally we formalized them as augmented finite state machines (Brooks 1986), although more recently we have thought of them as real-time rules (Brooks 1990*a*). They are too small to have a complete processor devoted to them in any machine beyond a CM-2, and even there the processors would be mostly idle. A better approach is to simulate parallelism in a single conventional processor with its own local memory.

For instance, Ferrell (1993) built a software system to control a 19 actuator six legged robot using about 60 of its sensors. She implemented it as more than 1500 parallel processes running on a single Phillips 68070. (It communicated with 7 peripheral processors which handled sensor data collection and 100Hz motor servoing.) Most of these parallel processes ran at rates varying between 10 and 25 Hertz. Each time each process ran, it took at most a few dozen instructions before blocking, waiting either for the passage of time or for some other process to send it a message. Clearly, low cost context switching was important.

The underlying computational model used on that robot—and with many tens of other autonomous mobile robots we have built—consisted of networks of message-passing augmented finite state machines. Each of these AFSMs was a separate process. The messages were sent over predefined 'wires' from a specific transmitting to a specific receiving AFSM. The messages were simple numbers (typically 8 bits) whose meaning depended on the designs of both the transmitter and the receiver. An AFSM had additional registers which held the most recent incoming message on any particular wire. This gives a very simple model of parallelism, even simpler than that of CSP (Hoare 1985). The registers could have their values fed into a local combinatorial circuit to produce new values for registers or to provide an output message. The network of AFSMs was totally asynchronous, but individual AFSMs could have fixed duration monostables which provided for dealing with the flow of time in the outside world. The behavioral competence of the system was improved by adding more behavior-specific network to the existing network. This process was called *layering*. This was a simplistic and crude analogy to evolutionary development. As with evolution, at every stage of the development the systems were tested. Each of the layers was a behavior-producing piece of network in its own right, although it might implicitly rely on the presence of earlier pieces of network. For instance, an *explore* layer did not need to explicitly avoid obstacles, as the designer knew that a previous *avoid* layer would take care of it. A fixed priority arbitration scheme was used to handle conflicts.

On top of the AFSM substrate we used another abstraction known as the Behavior Language, or BL (Brooks 1990*a*), which was much easier for the user to program with. The output of the BL compiler was a standard set of augmented finite state machines; by maintaining this compatibility all existing software could be retained. When programming in BL the user has complete access to full Common Lisp as a metalanguage by way of a macro mechanism. Thus the user could easily develop abstractions on top of BL, while still writing programs which compiled down to networks of AFSMs. In a sense, AFSMs played the role of assembly language in normal high level computer languages. But the structure of the AFSM networks enforced a programming style which naturally compiled into very efficient small processes. The structure of the Behavior Language enforced a modularity where data sharing was restricted to smallish sets of AFSMs, and whose only interfaces were essentially asynchronous 1-deep buffers.

In the humanoid project we believe much of the computation, especially for the lower levels of the system, will naturally be of a similar nature. We expect to perform different experiments where in some cases the higher level computations are of the same nature and in other cases the higher levels will be much more sym-

bolic in nature, although the symbolic bindings will be restricted to within individual processors. We need to use software and hardware environments which give support to these requirements without sacrificing the high levels of performance of which we wish to make use.

Software

For the software environment we have a number of requirements:

- There should be a good software development environment.
- The system should be completely portable over many hardware environments, so that we can upgrade to new parallel machines over the lifetime of this project.
- The system should provide efficient code for perceptual processing such as vision.
- The system should let us write high level symbolic programs when desired.
- The system language should be a standardized language that is widely known and understood.

In summary our software environment should let us gain easy access to high performance parallel computation.

We have chosen to use Common Lisp (Steele Jr. 1990) as the substrate for all software development. This gives us good programming environments including type checked debugging, rapid prototyping, symbolic computation, easy ways of writing embedded language abstractions, and automatic storage management. We believe that Common Lisp is superior to C (the other major contender) in all of these aspects.

The problem then is how to use Lisp in a massively parallel machine where each node may not have the vast amounts of memory that we have become accustomed to feeding Common Lisp implementations on standard Unix boxes.

We have a long history of building high performance Lisp compilers (Brooks, Gabriel & Steele Jr. 1982), including one of the two most common commercial Lisp compilers on the market; Lucid Lisp—Brooks, Posner, McDonald, White, Benson & Gabriel (1986).

Recently we have developed L (Brooks 1993), a retargetable small efficient Lisp which is a downwardly compatible subset of Common Lisp. When compiled for a 68000 based machine the load image (without the compiler) is only 140K bytes, but includes multiple values, strings, characters, arrays, a simplified but compatible package system, all the "ordinary" aspects of `format`, backquote and comma, `setf` etc., full Common Lisp lambda lists including optionals and keyword arguments, macros, an inspector, a debugger, `defstruct` (integrated with the inspector), `block`, `catch`, and `throw`, etc., full dynamic closures, a full lexical interpreter, floating point, fast garbage collection, and so on. The compiler runs in time linear in the size of an input expression, except in the presence of lexical closures. It nevertheless produces highly optimized code in most cases. L is missing `flet` and `labels`, generic arithmetic, bignums, rationals, complex numbers, the library of sequence functions (which can be written within L) and esoteric parts of `format` and packages.

The L system is an intellectual descendent of the dynamically retargetable Lucid Lisp compiler (Brooks et al. 1986) and the dynamically retargetable Behavior Language compiler (Brooks 1990*a*). The system is totally written in L with machine dependent backends for retargetting. The first backend is for the Motorola 68020 (and upwards) family, but it is easily retargeted to new architectures. The process consists of writing a simple machine description, providing code templates for about 100 primitive procedures (e.g., fixed precision integer `+`, `*`, `=`, etc., string indexing `CHAR` and other accessors, `CAR`, `CDR`, etc.), code macro expansion for about 20 pseudo instructions (e.g, procedure call, procedure exit, checking correct number of arguments, linking `CATCH` frames, etc.) and two corresponding sets of assembler routines which are too big to be expanded as code templates every time, but are so critical in speed that they need to be written in machine language, without the overhead of a procedure call, rather than in Lisp (e.g., `CONS`, spreading of multiple values on the stack, etc.). There is a version of the I/O system which operates by calling C routines (e.g., `fgetchar`, etc.; this is how the Macintosh version of L runs) so it is rather simple to port the system to any hardware platform we might choose to use in the future.

Note carefully the intention here: L is to be the delivery vehicle running on the brain hardware of the humanoid, potentially on hundreds or thousands of small processors. Since it is fully downward compatible with Common Lisp however, we can carry out code development and debugging on standard work stations with full programming environments (e.g., in Macintosh Common Lisp, or Lucid Common Lisp with Emacs 19 on a Unix box, or in the Harlequin programming environment on a Unix box). We can then dynamically link code into the running system on our parallel processors.

There are two remaining problems: (1) how to maintain super critical real-time performance when using a Lisp system without hard ephemeral garbage collection, and (2) how to get the level of within-processor parallelism described earlier.

The structure of L's implementation is such that multiple independent heaps can be maintained within a single address space, sharing all the code and data segments of the Lisp proper. In this way super-critical portions of a system can be placed in a heap where no consing is occurring, and hence there is no possibility that they will be blocked by garbage collection.

The Behavior Language (Brooks 1990*a*) is an example of a compiler which builds special purpose static schedulers for low overhead parallelism. Each process ran until blocked and the syntax of the language forced there to always be a blocking condition, so there was no need for pre-emptive scheduling. Additionally the syntax and semantics of the language guaranteed that there would be zero stack context needed to be saved when a blocking condition was reached. We will need to build a new scheduling system with L to address similar issues in this project. To fit in with the philosophy of the rest of the system it must be a dynamic scheduler so that new processes can be added and deleted as a user types to the Lisp listener of a particular processor. Reasonably straightforward data structures can keep these costs to manageable levels. It is rather straightforward to build a phase into the L compiler which can recognize the situations described above. Thus it is straightforward to implement a set of macros which will provide a language abstraction on top of Lisp which will provide all the functionality of the Behavior Language and which will additionally let us have dynamic scheduling. Almost certainly a pre-emptive scheduler will be needed in addition, as it would be difficult to enforce a computation time limit syntactically when Common Lisp will essentially be available to the programmer—at the very least the case of the pre-emptive scheduler having to strike down a process will be useful as a safety device, and will also act as a debugging tool for the user to identify time critical computations which are stressing the bounded computation style of writing. In other cases static analysis will be able to determine maximum stack requirements for a particular process, and so heap allocated stacks will be usable.[†]

The software system so far described will be used to implement crude forms of 'brain models', where computations will be organized in ways inspired by the sorts of anatomical divisions we see occurring in animal brains. Note that we are not saying we will build a model of a particular brain, but rather there will be a modularity inspired by such components as visual cortex, auditory cortex, etc., and within and across those components there will be further modularity, e.g., a particular subsystem to implement the vestibulo-ocular response (VOR).

Thus besides on-processor parallelism we will need to provide a modularity tool that packages processes into groups and limits data sharing between them. Each package will reside on a single processor, but often processors will host many such packages. A package that communicates with another package should be insulated at the syntax level from knowing whether the other package is on the same or a different processor. The communication medium between such packages will again be 1-deep buffers without queuing or receipt acknowledgment—any such acknowledgment will need to be implemented as a backward channel, much as we see throughout the cortex (Churchland & Sejnowski 1992). This packaging system can be implemented in Common Lisp as a macro package.

We expect all such system level software development to be completed in the first twelve months of the project.

Computational Hardware

The computational model presented in the previous section is somewhat different from that usually assumed in high performance parallel computer applications. Typically (Cypher, Ho, Konstantinidou & Messina 1993) there is a strong bias on system requirements from the sort of benchmarks that are used to evaluate performance. The standard benchmarks for modern high performance computation seem to be Fortran codes for hydrodynamics, molecular simulations, or graphics rendering. We are proposing a very different application with very different requirements; in particular we require real-time response to a wide variety of external and internal events, we require good symbolic computation performance, we require only integer rather than high performance floating point operations,[‡] we require delivery of messages only to specific sites determined at program design time, rather than at run-time, and we require the ability to do very fast context switches because of the large number of parallel processes that we intend to run on each individual processor.

The fact that we will not need to support pointer references across the computational substrate will mean that we can rely on much simpler, and therefore higher performance, parallel computers than many other researchers—we will not have to worry about a consistent global memory, cache coherence, or arbitrary message routing. Since these are different requirements than those that are normally considered, we have to make some measurements with actual programs before we can we can make an intelligent off the shelf choice of computer hardware.

In order to answer some of these questions we are currently building a zero-th generation parallel computer. It is being built on a very low budget with off the shelf components wherever possible (a few fairly simple printed circuit boards need to be fabricated). The processors are 16Mhz Motorola 68332s on a standard board built by Vesta Technology. These plug 16 to a backplane. The backplane provides each processor with six communications ports (using the integrated timing processor unit to generate the required signals

[†]The problem with heap allocated stacks in the general case is that there will be no overflow protection into the rest of heap.

[‡]Consider the dynamic range possible in single signal channels in the human brain and it soon becomes apparent that all that we wish to do is certainly achievable with neither span of 600 orders of magnitude, or 47 significant binary digits.

along with special chip select and standard address and data lines) and a peripheral processor port. The communications ports will be hand-wired with patch cables, building a fixed topology network. (The cables incorporate a single dual ported RAM (8K by 16 bits) that itself includes hardware semaphores writable and readable by the two processors being connected.) Background processes running on the 68332 operating system provide sustained rate transfers of 60Hz packets of 4K bytes on each port, with higher peak rates if desired. These sustained rates do consume processing cycles from the 68332. On non-vision processors we expect much lower rates will be needed, and even on vision processors we can probably reduce the packet frequency to around 15Hz. Each processor has an operating system, L, and the dynamic scheduler residing in 1M of EPROM. There is 1M of RAM for program, stack and heap space. Up to 256 processors can be connected together.

Up to 16 backplanes can be connected to a single front end processor (FEP) via a shared 500K baud serial line to a SCSI emulator. A large network of 68332s can span many FEPs if we choose to extend the construction of this zero-th prototype. Initially we will use a Macintosh as a FEP. Software written in Macintosh Common Lisp on the FEP will provide disk I/O services to the 68332's, monitor status and health packets from them, and provide the user with a Lisp listener to any processor they might choose.

The zero-th version uses the standard Motorola SPI (serial peripheral interface) to communicate with up to 16 Motorola 6811 processors per 68332. These are a single chip processor with onboard EEPROM (2K bytes) and RAM (256 bytes), including a timer system, an SPI interface, and 8 channels of analog to digital conversion. We are building a small custom board for this processor that includes opto-isolated motor drivers and some standard analog support for sensors[§].

We expect our first backplane to be operational by August 1st, 1993 so that we can commence experiments with our first prototype body. We will collect statistics on inter-processor communication throughput, effects of latency, and other measures so that we can better choose a larger scale parallel processor for more serious versions of the humanoid.

In the meantime, however, there are certain developments on the horizon within the MIT Artificial Intelligence Lab which we expect to capitalize upon in order to dramatically upgrade our computational systems for early vision, and hence the resolution at which we can afford to process images in real time. The first of these, expected in the fall will be a somewhat similar distributed processing system based on the much higher performance Texas Instrument C40, which comes with built in support for fixed topology message passing. We expect these systems to be available in the Fall '93 timeframe. In October '94 we expect to be able to make use of the Abacus system, a bit level reconfigurable vision front-end processor being built under ARPA sponsorship which promises Tera-op performance on 16 bit fixed precision operands. Both these systems will be simply integrable with our zero-th order parallel processor via the standard dual-ported RAM protocol that we are using.

[§]We currently have 28 operational robots in our labs each with between 3 and 5 of these 6811 processors, and several dozen other robots with at least 1 such processor on board. We have great experience in writing compiler backends for these processors (including BL) and great experience in using them for all sorts of servoing, sensor monitoring, and communications tasks.

Bodies

As with the computational hardware, we are also currently engaged in building a zero-th generation body for early experimentation and design refinement towards more serious constructions within the scope of this proposal. We are presently limited by budgetary constraints to building an immobile, armless, deaf, torso with only black and white vision.

In the following subsections we outline the constraints and requirements on a full scale humanoid body and also include where relevant details of our zero-th level prototype.

Eyes

There has been quite a lot of recent work on *animate vision* using saccading stereo cameras, most notably at Rochester (Ballard 1989), (Coombs 1992), but also more recently at many other institutions, such as Oxford University.

The humanoid needs a head with high mechanical performance eyeballs and foveated vision if it is to be able to participate in the world with people in a natural way. Even our earliest heads will include two eyes, with foveated vision, able to pan and tilt as a unit, and with independent saccading ability (three saccades per second) and vergence control of the eyes. Fundamental vision based behaviors will include a visually calibrated vestibular-ocular reflex, smooth pursuit, visually calibrated saccades, and object centered foveal relative depth stereo. Independent visual systems will provide peripheral and foveal motion cues, color discrimination, human face pop-outs, and eventually face recognition. Over the course of the project, object recognition based based on "representations" from body schemas and manipulation interactions will be developed. This is completely different from any conventional object recognition schemes, and can not be attempted without an integrated vision and manipulation environment as we propose.

The eyeballs need to be able to saccade up to about three times per second, stabilizing for 250ms at each stop. Additionally the yaw axes should be controllable for vergence to a common point and drivable in

a manner appropriate for smooth pursuit and for image stabilization as part of a vestibulo-ocular response (VOR) to head movement. The eyeballs do not need to be force or torque controlled but they do need good fast position and velocity control. We have previously built a single eyeball, *A-eye*, on which we implemented a model of VOR, ocular-kinetic response (OKR) and saccades, all of which used dynamic visually based calibration (Viola 1990).

Other active vision systems have had both eyeballs mounted on a single tilt axis. We will begin experiments with separate tilt axes but if we find that relative tilt motion is not very useful we will back off from this requirement in later versions of the head.

The cameras need to cover a wide field of view, preferably close to 180 degrees, while also giving a foveated central region. Ideally the images should be RGB (rather than the very poor color signal of standard NTSC). A resolution of 512 by 512 at both the coarse and fine scale is desirable.

Our zero-th version of the cameras are black and white only. Each eyeball consists of two small lightweight cameras mounted with parallel axes. One gives a 115 degree field of view and the other gives a 20 degree foveated region. In order to handle the images in real time in our zero-th parallel processor we will subsample the images to be much smaller than the ideal.

Later versions of the head will have full RGB color cameras, wider angles for the peripheral vision, much finer grain sampling of the images, and perhaps a colinear optics set up using optical fiber cables and beam splitters. With more sophisticated high speed processing available we will also be able to do experiments with log-polar image representations.

Ears, Voice

Almost no work has been done on sound understanding, as distinct from speech understanding. This project will start on sound understanding to provide a much more solid processing base for later work on speech input. Early behavior layers will spatially correlate noises with visual events, and spatial registration will be continuously self calibrating. Efforts will concentrate on using this physical cross-correlation as a basis for reliably pulling out interesting events from background noise, and mimicking the cocktail party effect of being able to focus attention on particular sound sources. Visual correlation with face pop-outs, etc., will then be used to be able to extract human sound streams. Work will proceed on using these sounds streams to mimic infant's abilities to ignore language dependent irrelevances. By the time we get to elementary speech we will therefore have a system able to work in noisy environments and accustomed to multiple speakers with varying accents.

Sound perception will consist of three high quality microphones. (Although the human head uses only two auditory inputs, it relies heavily on the shape of the external ear in determining the vertical component of directional sound source.) Sound generation will be accomplished using a speaker.

Sound is critical for several aspects of the robot's activity. First, sound provides immediate feedback for motor manipulation and positioning. Babies learn to find and use their hands by batting at and manipulating toys that jingle and rattle. Adults use such cues as contact noises—the sound of an object hitting the table—to provide feedback to motor systems. Second, sound aids in socialization even before the emergence of language. Patterns such as turn-taking and mimicry are critical parts of children's development, and adults use guttural gestures to express attitudes and other conversational cues. Certain signal tones indicate encouragement or disapproval to all ages and stages of development. Finally, even pre-verbal children use sound effectively to convey intent; until our robots develop true language, other sounds will necessarily be a major source of communication.

Torsos

In order for the humanoid to be able to participate in the same sorts of body metaphors as are used by humans, it needs to have a symmetric human-like torso. It needs to be able to experience imbalance, feel symmetry, learn to coordinate head and body motion for stable vision, and be able to experience relief when it relaxes its body. Additionally the torso must be able to support the head, the arms, and any objects they grasp.

The torsos we build will initially have a three degree of freedom hip, with the axes passing through a common point, capable of leaning and twisting to any position in about three seconds—somewhat slower than a human. The neck will also have three degrees of freedom, with the axes passing through a common point which will also lie along the spinal axis of the body. The head will be capable of yawing at 90 degrees per second—less than peak human speed, but well within the range of natural human motions. As we build later versions we expect to increase these performance figures to more closely match the abilities of a human.

Apart from the normal sorts of kinematic sensors, the torso needs a number of additional sensors specifically aimed at providing input fodder for the development of bodily metaphors. In particular, strain gauges on the spine can give the system a feel for its posture and the symmetry of a particular configuration, plus a little information about any additional load the torso might bear when an arm picks up something heavy. Heat sensors on the motors and the motor drivers will give feedback as to how much work has been done by the body recently, and current sensors on the motors will give an indication of how hard the system is working instantaneously.

Our zero-th level torso is roughly 18 inches from the

base of the spine to the base of the neck. This corresponds to a smallish adult. It uses DC motors with built in gearboxes. The main concern we have is how quiet it will be, as we do not want the sound perception system to be overwhelmed by body noise.

Later versions of the torsos will have touch sensors integrated around the body, will have more compliant motion, will be quieter, and will need to provide better cabling ducts so that the cables can all feed out through a lower body outlet.

Arms

The eventual manipulator system will be a compliant multi-degree of freedom arm with a rather simple hand. (A better hand would be nice, but hand research is not yet at a point where we can get an interesting, easy-to use, off-the-shelf hand.) The arm will be safe enough that humans can interact with it, handing it things and taking things from it. The arm will be compliant enough that the system will be able to explore its own body—for instance, by touching its head system—so that it will be able to develop its own body metaphors. The full design of the even the first pair of arms is not yet completely worked out, and current funding does not permit the inclusion of arms on the zero-th level humanoid. In this section, we describe our desiderata for the arms and hands.

We want the arms to be very compliant yet still able to lift weights of a few pounds so that they can interact with human artifacts in interesting ways. Additionally we want the arms to have redundant degrees of freedom (rather than the six seen in a standard commercial robot arm), so that in many circumstances we can 'burn' some of those degrees of freedom in order to align a single joint so that the joint coordinates and task coordinates very nearly match. This will greatly simplify control of manipulation. It is the sort of thing people do all the time: for example, when bracing an elbow or the base of the palm (or even their middle and last two fingers) on a table to stabilize the hand during some delicate (or not so delicate) manipulation.

The hands in the first instances will be quite simple; devices that can grasp from above relying heavily on mechanical compliance—they may have as few as one degree of control freedom.

More sophisticated, however, will be the sensing on the arms and hands. We will use forms of conductive rubber to get a sense of touch over the surface of the arm, so that it can detect (compliant) collisions it might participate in. As with the torso there will be liberal use of strain gauges, heat sensors and current sensors so that the system can have a 'feel' for how its arms are being used and how they are performing.

We also expect to move towards a more sophisticated type of hand in later years of this project. Initially, unfortunately, we will be forced to use motions of the upper joints of the arm for fine manipulation tasks. More sophisticated hands will allow us to use finger motions, with much lower inertias, to carry out these tasks.

Development Plan

We plan on modeling the brain at a level above the neural level, but below what would normally be thought of as the cognitive level.

We understand abstraction well enough to know how to engineer a system that has similar properties and connections to the human brain without having to model its detailed local wiring. At the same time it is clear from the literature that there is no agreement on how things are really organized computationally at higher or modular levels, or indeed whether it even makes sense to talk about modules of the brain (e.g., short term memory, and long term memory) as generative structures.

Nevertheless, we expect to be guided, or one might say inspired, by what is known about the high level connectivity within the human brain (although admittedly much of our knowledge actually comes from macaques and other primates and is only extrapolated to be true of humans, a problem of concern to some brain scientists (Crick & Jones 1993)). Thus for instance we expect to have identifiable clusters of processors which we will be able to point to and say they are performing a role similar to that of the cerebellum (e.g., refining gross motor commands into coordinated smooth motions), or the cortex (e.g., some aspects of searching generalization/specialization hierarchies in object recognition (Ullman 1991)).

At another level we will directly model human systems where they are known in some detail. For instance there is quite a lot known about the control of eye movements in humans (again mostly extrapolated from work with monkeys) and we will build in a vestibulo-ocular response (VOR), OKR, smooth pursuit, and saccades using the best evidence available on how this is organized in humans (Lisberger 1988).

A third level of modeling or inspiration that we will use is at the developmental level. For instance once we have some sound understanding developed, we will use models of what happens in child language development to explore ways of connecting physical actions in the world to a ground of language and the development of symbols (Bates 1979), (Bates, Bretherton & Snyder 1988), including indexical (Lempert & Kinsbourne 1985) and turn-taking behavior, interpretation of tone and facial expressions and the early use of memorized phrases.

Since we will have a number of faculty, post-doctoral fellows, and graduate students working on concurrent research projects, and since we will have a number of concurrently active humanoid robots, not all pieces that are developed will be intended to fit together exactly. Some will be incompatible experiments in alternate ways of building subsystems, or putting them together. Some will be pushing on particular issues in

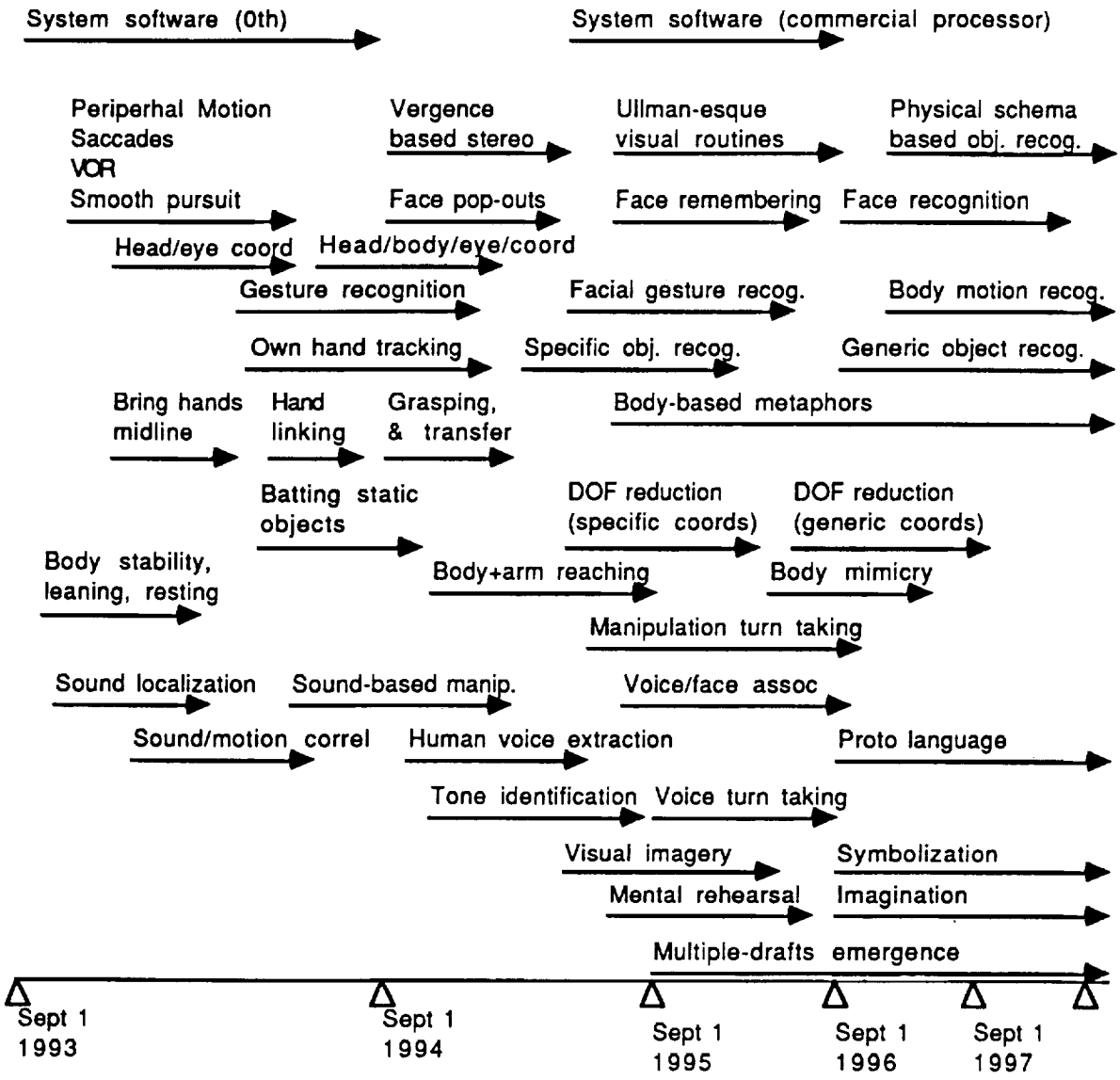
System software (0th)
System software (commercial processor)
Peripherhal Motion
Saccades
VOR
Smooth pursuit
Vergence
based stereo
Face pop-outs
Ullman-esque
visual routines
Face remembering
Physical schema
based obj. recog.
Face recognition
Head/eye coord
Head/body/eye/coord
Gesture recognition
Facial gesture recog.
Body motion recog.
Own hand tracking
Specific obj. recog.
Generic object recog.
Bring hands
midline
Hand
linking
Grasping,
& transfer
Body-based metaphors
Batting static
objects
DOF reduction
(specific coords)
DOF reduction
(generic coords)
Body stability,
leaning, resting
Body+arm reaching
Body mimicry
Manipulation turn taking
Sound localization
Sound-based manip.
Voice/face assoc
Sound/motion correl
Human voice extraction
Proto language
Tone identification
Voice turn taking
Visual imagery
Symbolization
Mental rehearsal
Imagination
Multiple-drafts emergence
Sept 1
1993
Sept 1
1994
Sept 1
1995
Sept 1
1996
Sept 1
1997

Figure 1

language, say, that may not be very related to some particular other issues, e.g., saccades. Also, quite clearly, at this stage we can not have a development plan fully worked out for five years, as many of the early results will change the way we think about the problems and what should be the next steps.

In figure 1, we summarize our current plans for developing software systems on board our series of humanoids. In many cases there will be earlier work off-board the robots, but to keep clutter down in the diagram we have omitted that work here.

Acknowledgements

This paper has benefitted from conversations with and comments by Catherine Harris, Dan Dennett, Marcel Kinsbourne. We are also indebted to the members of our research groups, individually and collectively, who have shared their thoughts and enthusiasm.

References

Agre, P. E. & Chapman, D. (1987), Pengi: An Implementation of a Theory of Activity, in 'Proceedings of the Sixth National Conference on Artificial Intelligence', Morgan Kaufmann, Seattle, Washington, pp. 196–201.

Allen, J., Hendler, J. & Tate, A., eds (1990), Readings in Planning, Morgan Kaufmann, Los Altos, California.

Angle, C. M. & Brooks, R. A. (1990), Small Planetary Rovers, in 'IEEE/RSJ International Workshop on Intelligent Robots and Systems', Ikabara, Japan, pp. 383–388.

Arbib, M. A. (1964), Brains, Machines and Mathematics, McGraw-Hill, New York, New York.

Ashby, W. R. (1956), An Introduction to Cybernetics, Chapman and Hall, London, United Kingdom.

Ballard, D. H. (1989), Reference Frames for Active Vision, in 'Proceedings of the International Joint Conference on Artificial Intelligence', Detroit, Michigan, pp. 1635–1641.

Bates, E. (1979), The Emergence of Symbols, Academic Press, New York, New York.

Bates, E., Bretherton, I. & Snyder, L. (1988), From First Words to Grammar, Cambridge University Press, Cambridge, United Kingdom.

Beer, R. D. (1990), Intelligence as Adaptive Behavior, Academic Press, Cambridge, Massachusetts.

Berkeley, E. C. (1949), Giant Brains or Machines that Think, John Wiley & Sons, New York, New York.

Bickhard, M. H. (n.d.), How to Build a Machine with Emergent Representational Content, Unpublished manuscript, University of Texas, Austin.

Brachman, R. J. & Levesque, H. J., eds (1985), Readings in Knowledge Representation, Morgan Kaufmann, Los Altos, California.

Braddick, O., Atkison, J., Hood, B., Harkness, W. & an Faraneh Vargha-Khadem, G. J. (1992), 'Possible blindsight in infants lacking one cerebral hemisphere', Nature **360**, 461–463.

Brooks, R. A. (1986), 'A Robust Layered Control System for a Mobile Robot', IEEE Journal of Robotics and Automation **RA-2**, 14–23.

Brooks, R. A. (1989), 'A Robot That Walks: Emergent Behavior from a Carefully Evolved Network', Neural Computation 1(2), 253–262.

Brooks, R. A. (1990a), The Behavior Language User's Guide, Memo 1227, Massachusetts Institute of TechnologyArtificial Intelligence Lab, Cambridge, Massachusetts.

Brooks, R. A. (1990b), Elephants Don't Play Chess, in P. Maes, ed., 'Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back', MIT Press, Cambridge, Massachusetts, pp. 3–15.

Brooks, R. A. (1991a), Intelligence Without Reason, in 'Proceedings of the 1991 International Joint Conference on Artificial Intelligence', pp. 569–595.

Brooks, R. A. (1991b), 'New Approaches to Robotics', Science **253**, 1227–1232.

Brooks, R. A. (1993), L: A Subset of Common Lisp, Technical report, Massachusetts Institute of TechnologyArtificial Intelligence Lab.

Brooks, R. A., Gabriel, R. P. & Steele Jr., G. L. (1982), An Optimizing Compiler for Lexically Scoped Lisp, in 'Proceedings of the 1982 Symposium on Compiler Construction. ACM SIGPLAN', Boston, Massachusetts, pp. 261–275. Publised as ACM SIGPLAN Notices 17, 6 (June 1982).

Brooks, R. A., Posner, D. B., McDonald, J. L., White, J. L., Benson, E. & Gabriel, R. P. (1986), Design of An Optimizing Dynamically Retargetable Compiler for Common Lisp, in 'Proceedings of the 1986 ACM Symposium on Lisp and Functional Programming', Cambridge, Massachusetts, pp. 67–85.

Caudill, M. (1992), In Our Own Image: Building An Artificial Person, Oxford University Press, New York, New York.

Churchland, P. S. & Sejnowski, T. J. (1992), The Computational Brain, MIT Press, Cambridge, Massachusetts.

Connell, J. H. (1987), Creature Building with the Subsumption Architecture, in 'Proceedings of the International Joint Conference on Artificial Intelligence', Milan, Italy, pp. 1124–1126.

Connell, J. H. (1990), Minimalist Mobile Robotics: A Colony-style Architecture for a Mobile Robot, Academic Press, Cambridge, Massachusetts. also MIT TR-1151.

Coombs, D. J. (1992), Real-time Gaze Holding in Binocular Robot Vision, PhD thesis, University of Rochester, Department of CS, Rochester, New York.

Crick, F. & Jones, E. (1993), 'Backwardness of human neuroanatomy', Nature **361**, 109–110.

Cypher, R., Ho, A., Konstantinidou, S. & Messina, P. (1993), Architectural Requirements of Parallel Scientific Applications with Explicit Communication, in 'IEEE Proceedings of the 20th International Symposium on Computer Architecture', San Diego, California, pp. 2–13.

Damasio, H. & Damasio, A. R. (1989), Lesion Analysis in Neuropsychology, Oxford University Press, New York, New York.

Dennett, D. C. (1991), Consciousness Explained, LIttle, Brown, Boston, Massachusetts.

Dennett, D. C. & Kinsbourne, M. (1992), 'Time and the Observer: The Where and When of Consciousness in the Brain', Brain and Behavioral Sciences **15**, 183–247.

Drescher, G. L. (1991), Made-Up Minds: A Constructivist Approach to Artificial Intelligence, MIT Press, Cambridge, Massachusetts.

Edelman, G. M. (1987), Neural Darwinsim: The Theory of Neuronal Group Selection, Basic Books, New York, New York.

Edelman, G. M. (1989), The Remembered Present: A Biological Theory of Consciousness, Basic Books, New York, New York.

Edelman, G. M. (1992), Bright Air, Brilliant Fire: On the Matter of Mind, Basic Books, New York, New York.

Fendrich, R., Wessinger, C. M. & Gazzaniga, M. S. (1992), 'Residual Vision in a Scotoma: Implications for Blindsight', Science **258**, 1489–1491.

Ferrell, C. (1993), Robust Agent Control of an Autonomous Robot with Many Sensors and Actuators, Master's thesis, MIT, Department of EECS, Cambridge, Massachusetts.

Fodor, J. A. (1983), The Modularity of Mind, Bradford Books, MIT Press, Cambridge, Massachusetts.

Harris, C. L. (1991), Parallel Distributed Processing Models and Metaphors for Language and Development, PhD thesis, University of California, Department of Cognitive Science, San Diego, California.

Haugeland, J. (1985), Artificial Intelligence: The Very Idea, MIT Press, Cambridge, Massachusets.

Hoare, C. A. R. (1985), Communicating Sequential Processes, Prentice-Hall, Englewood Cliffs, New Jersey.

Hobbs, J. & Moore, R., eds (1985), Formal Theories of the Commonsense World, Ablex Publishing Co., Norwood, New Jersey.

Horswill, I. D. (1993), Specialization of Perceptual Processes, PhD thesis, MIT, Department of EECS, Cambridge, Massachusetts.

Horswill, I. D. & Brooks, R. A. (1988), Situated Vision in a Dynamic World: Chasing Objects, in 'Proceedings of the Seventh Annual Meeting of the American Association for Artificial Intelligence', St. Paul, Minnesota, pp. 796–800.

Johnson, M. (1987), The Body In The Mind, University of Chicago Press, Chicago, Illinois.

Kinsbourne, M. (1987), Mechanisms of unilateral neglect, in M. Jeannerod, ed., 'Neurophysiological and Neuropsychological Aspects of Spatial Neglect', Elsevier, North Holland.

Kinsbourne, M. (1988), Integrated field theory of consciousness, in A. Marcel & E. Bisiach, eds, 'The Concept of Consciousness in Contemporary Science', Oxford University Press, London, England.

Kosslyn, S. (1993), Image and brain: The resolution of the imagery debate, Harvard University Press, Cambridge, Massachusetts.

Kuipers, B. & Byun, Y.-T. (1991), 'A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations', Robotics and Autonomous Systems **8**, 47–63.

Lakoff, G. (1987), Women, Fire, and Dangerous Things, University of Chicago Press, Chicago, Illinois.

Lakoff, G. & Johnson, M. (1980), Metaphors We Live By, University of Chicago Press, Chicago, Illinois.

Langacker, R. W. (1987), Foundations of cognitive grammar, Volume 1, Stanford University Press, Palo Alto, California.

Lempert, H. & Kinsbourne, M. (1985), 'Possible origin of speech in selective orienting', Psychological Bulletin **97**, 62–73.

Lisberger, S. G. (1988), 'The neural basis for motor learning in the vestibulo-ocular reflex in monkeys', Trends in Neuroscience **11**, 147–152.

Marr, D. (1982), Vision, W. H. Freeman, San Francisco, California.

Mataric, M. J. (1992a), Designing Emergent Behaviors: From Local Interactions to Collective Intelligence, in 'Proceedings of the Second International Conference on Simulation of Adaptive Behavior', MIT Press, Cambridge, Massachusetts, pp. 432–441.

Mataric, M. J. (1992b), 'Integration of Representation Into Goal-Driven Behavior-Based Robots', IEEE Journal of Robotics and Automation **8**(3), 304–312.

McCarthy, R. A. & Warrington, E. K. (1988), 'Evidence for Modality-Specific Systems in the Brain', Nature **334**, 428–430.

McCarthy, R. A. & Warrington, E. K. (1990), Cognitive Neuropsychology, Academic Press, San Diego, California.

Minsky, M. (1986), The Society of Mind, Simon and Schuster, New York, New York.

Minsky, M. & Papert, S. (1969), Perceptrons, MIT Press, Cambridge, Massachusetts.

Newcombe, F. & Ratcliff, G. (1989), Disorders of Visuspatial Analysis, in 'Handbook of Neuropsychology, Volume 2', Elsevier, New York, New York.

Newell, A. & Simon, H. A. (1981), Computer Science as Empirical Inquiry: Symbols and Search, in J. Haugeland, ed., 'Mind Design', MIT Press, Cambridge, Massachusetts, chapter 1, pp. 35–66.

Penrose, R. (1989), The Emporer's New Mind, Oxford University Press, Oxford, United Kingdom.

Philip Teitelbaum, V. C. P. & Pellis, S. M. (1990), Can Allied Reflexes Promote the Integration of a Robot's Behavior, in 'Proceedings of the First International Conference on Simulation of Adaptive Behavior', MIT Press, Cambridge, Massachusetts, pp. 97–104.

Pomerleau, D. A. (1991), 'Efficient Training of Artificial Neural Networks for Autonomous Navigation', Neural Computation.

Rosenblatt, F. (1962), Principles of Neurodynamics, Spartan, New York, New York.

Rosenschein, S. J. & Kaelbling, L. P. (1986), The Synthesis of Digital Machines with Provable Epestemic Properties, in J. Y. Halpern, ed., 'Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge', Morgan Kaufmann, Monterey, California, pp. 83–98.

Rumelhart, D. E. & McClelland, J. L., eds (1986), Parallel Distributed Processing, MIT Press, Cambridge, Massachusetts.

Searle, J. R. (1992), The Rediscovery of the Mind, MIT Press, Cambridge, Massachusetts.

Simon, H. A. (1969), The Sciences of the Artificial, MIT Press, Cambridge, Massachusetts.

Springer, S. P. & Deutsch, G. (1981), Left Brain, Right Brain, W.H. Freeman and Company, New York.

Steele Jr., G. L. (1990), Common Lisp, The Language, second edn, Digital Press.

Stein, L. A. (to appear), 'Imagination and Situated Cognition', Journal of Experimental and Theoretical Artificial Intelligence.

Turing, A. M. (1970), Intelligent Machinery, in B. Meltzer & D. Michie, eds, 'Machine Intelligence 5', American Elsevier Publishing, New York, New York, pp. 3–23.

Ullman, S. (1991), Sequence-Seeking and Counter Streams: A Model for Information Processing in the Cortex, Memo 1311, Massachusetts Institute of TechnologyArtificial Intelligence Lab, Cambridge, Massachusetts.

Viola, P. A. (1990), Adaptive Gaze Control, Master's thesis, MIT, Department of EECS, Cambridge, Massachusetts.

Weiskrantz, L. (1986), Blindsight, Oxford University Press, Oxford, United Kingdom.

Yanco, H. & Stein, L. A. (1993), An Adaptive Communication Protocol for Cooperating Mobile Robots, in J.-A. Meyer, H. Roitblat & S. Wilson, eds, 'From Animals to Animats: Proceedings of the Second Conference on the Simulation of Adaptive Behavior', MIT Press, Cambridge, Massachusetts, pp. 478–485.