Outline for a Theory of Intelligence

James S. Albus

Abstract—**Intelligence is defined as that which produces successful behavior. Intelligence is assumed to result from natural selection. A model is proposed that integrates knowledge from research in both natural and artificial systems. The model consists of a hierarchical system architecture wherein: 1) control bandwidth decreases about an order of magnitude at each higher level, 2) perceptual resolution of spatial and temporal patterns contracts about an order-of-magnitude at each higher level, 3) goals expand in scope and planning horizons expand in space and time about an order-of-magnitude at each higher level, and 4) models of the world and memories of events expand their range in space and time by about an order-of-magnitude at each higher level. At each level, functional modules perform behavior generation (task decomposition planning and execution), world modeling, sensory processing, and value judgment. Sensory feedback control loops are closed at every level.**

I. Introduction

MUCH IS UNKNOWN about intelligence, and much will remain beyond human comprehension for a very long time. The fundamental nature of intelligence is only dimly understood, and the elements of self consciousness, perception, reason, emotion, and intuition are cloaked in mystery that shrouds the human psyche and fades into the religious. Even the definition of intelligence remains a subject of controversy, and so must any theory that attempts to explain what intelligence is, how it originated, or what are the fundamental processes by which it functions.

Yet, much is known, both about the mechanisms and function of intelligence. The study of intelligent machines and the neurosciences are both extremely active fields. Many millions of dollars per year are now being spent in Europe, Japan, and the United States on computer integrated manufacturing, robotics, and intelligent machines for a wide variety of military and commercial applications. Around the world, researchers in the neurosciences are searching for the anatomical, physiological, and chemical basis of behavior.

Neuroanatomy has produced extensive maps of the interconnecting pathways making up the structure of the brain. Neurophysiology is demonstrating how neurons compute functions and communicate information. Neuropharmacology is discovering many of the transmitter substances that modify value judgments, compute reward and punishment, activate behavior, and produce learning. Psychophysics provides many clues as to how individuals perceive objects, events, time, and space, and how they reason about relationships between themselves and the external world. Behavioral psychology adds information about mental development, emotions, and behavior.

Research in learning automata, neural nets, and brain modeling has given insight into learning and the similarities and differences between neuronal and electronic computing processes. Computer science and artificial intelligence is probing the nature of language and image understanding, and has made significant progress in rule based reasoning, planning, and problem solving. Game theory and operations research have developed methods for decision making in the face of uncertainty. Robotics and autonomous vehicle research has produced advances in real-time sensory processing, world modeling, navigation, trajectory generation, and obstacle avoidance. Research in automated manufacturing and process control has produced intelligent hierarchical controls, distributed databases, representations of object geometry and material properties, data driven task sequencing, network communications, and multiprocessor operating systems. Modern control theory has developed precise understanding of stability, adaptability, and controllability under various conditions of feedback and noise. Research in sonar, radar, and optical signal processing has developed methods for fusing sensory input from multiple sources, and assessing the believability of noisy data.

Progress is rapid, and there exists an enormous and rapidly growing literature in each of the previous fields. What is lacking is a general theoretical model of intelligence that ties all these separate areas of knowledge into a unified framework. This paper is an attempt to formulate at least the broad outlines of such a model.

The ultimate goal is a general theory of intelligence that encompasses both biological and machine instantiations. The model presented here incorporates knowledge gained from many different sources and the discussion frequently shifts back and forth between natural and artificial systems. For example, the definition of intelligence in Section II addresses both natural and artificial systems. Section III treats the origin and function of intelligence from the standpoint of biological evolution. In Section IV, both natural and artificial system elements are discussed. The system architecture described in Sections V–VII derives almost entirely from research in robotics and control theory for devices ranging from undersea vehicles to automatic factories. Sections VIII–XI on behavior generation, Sections XII and XIII on world modeling, and Section XIV on sensory processing are elaborations of the system architecture of Section V–VII. These sections all contain numerous references to neurophysiological, psychological, and psychophysical phenomena that support the model, and frequent analogies are drawn between biological and artificial

Manuscript received March 16, 1990; revised November 16, 1990.

The author is with the Robot Systems Division Center for Manufacturing Engineering, National Institute of Standards and Technology, Gaithersburg, MD 20899.

IEEE Log Number 9042583.

0018-9472/91/0500-0473$01.00 © 1991 IEEE

systems. The value judgments, described in Section XV, are mostly based on the neurophysiology of the limbic system and the psychology of emotion. Section XVI on neural computation and Section XVII on learning derive mostly from neural net research.

The model is described in terms of definitions, axioms, theorems, hypotheses, conjectures, and arguments in support of them. Axioms are statements that are assumed to be true without proof. Theorems are statements that the author feels could be demonstrated true by existing logical methods or empirical evidence. Few of the theorems are proven, but each is followed by informal discussions that support the theorem and suggest arguments upon which a formal proof might be constructed. Hypotheses are statements that the author feels probably could be demonstrated through future research. Conjectures are statements that the author feels might be demonstrable.

II. Definition Of Intelligence

In order to be useful in the quest for a general theory, the definition of intelligence must not be limited to behavior that is not understood. A useful definition of intelligence should span a wide range of capabilities, from those that are well understood, to those that are beyond comprehension. It should include both biological and machine embodiments, and these should span an intellectual range from that of an insect to that of an Einstein, from that of a thermostat to that of the most sophisticated computer system that could ever be built. The definition of intelligence should, for example, include the ability of a robot to spotweld an automobile body, the ability of a bee to navigate in a field of wild flowers, a squirrel to jump from limb to limb, a duck to land in a high wind, and a swallow to work a field of insects. It should include what enables a pair of blue jays to battle in the branches for a nesting site, a pride of lions to pull down a wildebeest, a flock of geese to migrate south in the winter. It should include what enables a human to bake a cake, play the violin, read a book, write a poem, fight a war, or invent a computer.

At a minimum, intelligence requires the ability to sense the environment, to make decisions, and to control action. Higher levels of intelligence may include the ability to recognize objects and events, to represent knowledge in a world model, and to reason about and plan for the future. In advanced forms, intelligence provides the capacity to perceive and understand, to choose wisely, and to act successfully under a large variety of circumstances so as to survive, prosper, and reproduce in a complex and often hostile environment.

From the viewpoint of control theory, intelligence might be defined as a knowledgeable "helmsman of behavior". Intelligence is the integration of knowledge and feedback into a sensory-interactive goal-directed control system that can make plans, and generate effective, purposeful action directed toward achieving them.

From the viewpoint of psychology, intelligence might be defined as a behavioral strategy that gives each individual a means for maximizing the likelihood of propagating its own genes. Intelligence is the integration of perception, reason, emotion, and behavior in a sensing, perceiving, knowing, caring, planning, acting system that can succeed in achieving its goals in the world.

For the purposes of this paper, intelligence will be defined as the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal.

Both the criteria of success and the systems ultimate goal are defined external to the intelligent system. For an intelligent machine system, the goals and success criteria are typically defined by designers, programmers, and operators. For intelligent biological creatures, the ultimate goal is gene propagation, and success criteria are defined by the processes of natural selection.

Theorem: There are degrees, or levels, of intelligence, and these are determined by: 1) the computational power of the system's brain (or computer), 2) the sophistication of algorithms the system uses for sensory processing, world modeling, behavior generating, value judgment, and global communication, and 3) the information and values the system has stored in its memory.

Intelligence can be observed to grow and evolve, both through growth in computational power, and through accumulation of knowledge of how to sense, decide, and act in a complex and changing world. In artificial systems, growth in computational power and accumulation of knowledge derives mostly from human hardware engineers and software programmers. In natural systems, intelligence grows, over the lifetime of an individual, through maturation and learning; and over intervals spanning generations, through evolution.

Note that learning is not required in order to be intelligent, only to become more intelligent as a result of experience. Learning is defined as consolidating short-term memory into long-term memory, and exhibiting altered behavior because of what was remembered. In Section X, learning is discussed as a mechanism for storing knowledge about the external world, and for acquiring skills and knowledge of how to act. It is, however, assumed that many creatures can exhibit intelligent behavior using instinct, without having learned anything.

III. The Origin And Function Of Intelligence

Theorem: Natural intelligence, like the brain in which it appears, is a result of the process of natural selection.

The brain is first and foremost a control system. Its primary function is to produce successful goal-seeking behavior in finding food, avoiding danger, competing for territory, attracting sexual partners, and caring for offspring. All brains that ever existed, even those of the tiniest insects, generate and control behavior. Some brains produce only simple forms of behavior, while others produce very complex behaviors. Only the most recent and highly developed brains show any evidence of abstract thought.

Theorem: For each individual, intelligence provides a mechanism for generating biologically advantageous behavior.

Intelligence improves an individual's ability to act effectively and choose wisely between alternative behaviors. All else being equal, a more intelligent individual has many advantages over less intelligent rivals in acquiring choice territory, gaining access to food, and attracting more desirable breeding partners. The intelligent use of aggression helps to improve an individual's position in the social dominance hierarchy. Intelligent predation improves success in capturing prey. Intelligent exploration improves success in hunting and establishing territory. Intelligent use of stealth gives a predator the advantage of surprise. Intelligent use of deception improves the prey's chances of escaping from danger.

Higher levels of intelligence produce capabilities in the individual for thinking ahead, planning before acting, and reasoning about the probable results of alternative actions. These abilities give to the more intelligent individual a competitive advantage over the less intelligent in the competition for survival and gene propagation. Intellectual capacities and behavioral skills that produce successful hunting and gathering of food, acquisition and defense of territory, avoidance and escape from danger, and bearing and raising offspring tend to be passed on to succeeding generations. Intellectual capabilities that produce less successful behaviors reduce the survival probability of the brains that generate them. Competition between individuals thus drives the evolution of intelligence within a species.

Theorem: For groups of individuals, intelligence provides a mechanism for cooperatively generating biologically advantageous behavior.

The intellectual capacity to simply congregate into flocks, herds, schools, and packs increases the number of sensors watching for danger. The ability to communicate danger signals improves the survival probability of all individuals in the group. Communication is most advantageous to those individuals who are the quickest and most discriminating in the recognition of danger messages, and most effective in responding with appropriate action. The intelligence to cooperate in mutually beneficial activities such as hunting and group defense increases the probability of gene propagation for all members of the group.

All else being equal, the most intelligent individuals and groups within a species will tend to occupy the best territory, be the most successful in social competition, and have the best chances for their offspring surviving. All else being equal, more intelligent individuals and groups will win out in serious competition with less intelligent individuals and groups.

Intelligence is, therefore, the product of continuous competitive struggles for survival and gene propagation that has taken place between billions of brains, over millions of years. The results of those struggles have been determined in large measure by the intelligence of the competitors.

A. *Communication and Language*

Definition: Communication is the transmission of information between intelligent systems.

Definition: Language is the means by which information is encoded for purposes of communication.

Language has three basic components: vocabulary, syntax, and semantics. Vocabulary is the set of words in the language. Words may be represented by symbols. Syntax, or grammar, is the set of rules for generating strings of symbols that form sentences. Semantics is the encoding of information into meaningful patterns, or messages. Messages are sentences that convey useful information.

Communication requires that information be: 1) encoded, 2) transmitted, 3) received, 4) decoded, and 5) understood. Understanding implies that the information in the message has been correctly decoded and incorporated into the world model of the receiver.

Communication may be either intentional or unintentional. Intentional communication occurs as the result of a sender executing a task whose goal it is to alter the knowledge or behavior of the receiver to the benefit of the sender. Unintentional communication occurs when a message is unintentionally sent, or when an intended message is received and understood by someone other than the intended receiver. Preventing an enemy from receiving and understanding communication between friendly agents can often be crucial to survival.

Communication and language are by no means unique to human beings. Virtually all creatures, even insects, communicate in some way, and hence have some form of language. For example, many insects transmit messages announcing their identity and position. This may be done acoustically, by smell, or by some visually detectable display. The goal may be to attract a mate, or to facilitate recognition and/or location by other members of a group. Species of lower intelligence, such as insects, have very little information to communicate, and hence have languages with only a few of what might be called words, with little or no grammar. In many cases, language vocabularies include motions and gestures (i.e., body or sign language) as well as acoustic signals generated by variety of mechanisms from stamping the feet, to snorts, squeals, chirps, cries, and shouts.

Theorem: In any species, language evolves to support the complexity of messages that can be generated by the intelligence of that species.

Depending on its complexity, a language may be capable of communicating many messages, or only a few. More intelligent individuals have a larger vocabulary, and are quicker to understand and act on the meaning of messages.

Theorem: To the receiver, the benefit, or value, of communication is roughly proportional to the product of the amount of information contained in the message, multiplied by the ability of the receiver to understand and act on that information, multiplied by the importance of the act to survival and gene propagation of the receiver. To the sender, the benefit is the value of the receiver's action to the sender, minus the danger incurred by transmitting a message that may be intercepted by, and give advantage to, an enemy.

Greater intelligence enhances both the individual's and the group's ability to analyze the environment, to encode and transmit information about it, to detect messages, to recognize their significance, and act effectively on information received. Greater intelligence produces more complex languages capable of expressing more information, i.e., more messages with more shades of meaning.

In social species, communication also provides the basis for societal organization. Communication of threats that warn of aggression can help to establish the social dominance hierarchy, and reduce the incidence of physical harm from fights over food, territory, and sexual partners. Communication of alarm signals indicate the presence of danger, and in some cases, identify its type and location. Communication of pleas for help enables group members to solicit assistance from one another. Communication between members of a hunting pack enable them to remain in formation while spread far apart, and hence to hunt more effectively by cooperating as a team in the tracking and killing of prey.

Among humans, primitive forms of communication include facial expressions, cries, gestures, body language, and pantomime. The human brain is, however, capable of generating ideas of much greater complexity and subtlety than can be expressed through cries and gestures. In order to transmit messages commensurate with the complexity of human thought, human languages have evolved grammatical and semantic rules capable of stringing words from vocabularies consisting of thousands of entries into sentences that express ideas and concepts with exquisitely subtle nuances of meaning. To support this process, the human vocal apparatus has evolved complex mechanisms for making a large variety of sounds.

B. *Human Intelligence and Technology*

Superior intelligence alone made man a successful hunter. The intellectual capacity to make and use tools, weapons, and spoken language made him the most successful of all predators. In recent millennia, human levels of intelligence have led to the use of fire, the domestication of animals, the development of agriculture, the rise of civilization, the invention of writing, the building of cities, the practice of war, the emergence of science, and the growth of industry. These capabilities have extremely high gene propagation value for the individuals and societies that possess them relative to those who do not. Intelligence has thus made modern civilized humans the dominant species on the planet Earth.

For an individual human, superior intelligence is an asset in competing for position in the social dominance hierarchy. It conveys advantage for attracting and winning a desirable mate, in raising a large, healthy, and prosperous family, and seeing to it that one's offspring are well provided for. In competition between human groups, more intelligent customs and traditions, and more highly developed institutions and technology, lead to the dominance of culture and growth of military and political power. Less intelligent customs, traditions, and practices, and less developed institutions and technology, lead to economic and political decline and eventually to the demise of tribes, nations, and civilizations.

IV. The Elements Of Intelligence

Theorem: There are four system elements of intelligence: sensory processing, world modeling, behavior generation, and value judgment. Input to, and output from, intelligent systems are via sensors and actuators.

1) Actuators: Output from an intelligent system is produced by actuators that move, exert forces, and position arms, legs, hands, and eyes. Actuators generate forces to point sensors, excite transducers, move manipulators, handle tools, steer and propel locomotion. An intelligent system may have tens, hundreds, thousands, even millions of actuators, all of which must be coordinated in order to perform tasks and accomplish goals. Natural actuators are muscles and glands. Machine actuators are motors, pistons, valves, solenoids, and transducers.

2) Sensors: Input to an intelligent system is produced by sensors, which may include visual brightness and color sensors; tactile, force, torque, position detectors; velocity, vibration, acoustic, range, smell, taste, pressure, and temperature measuring devices. Sensors may be used to monitor both the state of the external world and the internal state of the intelligent system itself. Sensors provide input to a sensory processing system.

3) Sensory Processing: Perception takes place in a sensory processing system element that compares sensory observations with expectations generated by an internal world model. Sensory processing algorithms integrate similarities and differences between observations and expectations over time and space so as to detect events and recognize features, objects, and relationships in the world. Sensory input data from a wide variety of sensors over extended periods of time are fused into a consistent unified perception of the state of the world. Sensory processing algorithms compute distance, shape, orientation, surface characteristics, physical and dynamical attributes of objects and regions of space. Sensory processing may include recognition of speech and interpretation of language and music.

4) World Model: The world model is the intelligent system's best estimate of the state of the world. The world model includes a database of knowledge about the world, plus a database management system that stores and retrieves information. The world model also contains a simulation capability that generates expectations and predictions. The world model thus can provide answers to requests for information about the present, past, and probable future states of the world. The world model provides this information service to the behavior generation system element, so that it can make intelligent plans and behavioral choices, to the sensory processing system element, in order for it to perform correlation, model matching, and model based recognition of states, objects, and events, and to the value judgment system element in order for it to compute values such as cost, benefit, risk, uncertainty, importance, attractiveness, etc. The world model is kept up-to-date by the sensory processing system element.

5) Value Judgment: The value judgment system element determines what is good and bad, rewarding and punishing, important and trivial, certain and improbable. The value judgment system evaluates both the observed state of the world and the predicted results of hypothesized plans. It computes costs, risks, and benefits both of observed situations and of planned activities. It computes the probability of correctness and assigns believability and uncertainty parameters to state variables. It also assigns attractiveness, or repulsiveness to objects, events, regions of space, and other creatures. The value judgment system thus provides the basis for making decisions—for choosing one action as opposed to another, or for pursuing one object and fleeing from another. Without value judgments, any biological creature would soon be eaten by others, and any artificially intelligent system would soon be disabled by its own inappropriate actions.

6) Behavior Generation: Behavior results from a behavior generating system element that selects goals, and plans and executes tasks. Tasks are recursively decomposed into subtasks, and subtasks are sequenced so as to achieve goals. Goals are selected and plans generated by a looping interaction between behavior generation, world modeling, and value judgment elements. The behavior generating system hypothesizes plans, the world model predicts the results of those plans, and the value judgment element evaluates those results. The behavior generating system then selects the plans with the highest evaluations for execution. The behavior generating system element also monitors the execution of plans, and modifies existing plans whenever the situation requires.

Each of the system elements of intelligence are reasonably well understood. The phenomena of intelligence, however, requires more than a set of disconnected elements. Intelligence requires an interconnecting system architecture that enables the various system elements to interact and communicate with each other in intimate and sophisticated ways.

A system architecture is what partitions the system elements of intelligence into computational modules, and interconnects the modules in networks and hierarchies. It is what enables the behavior generation elements to direct sensors, and to focus sensory processing algorithms on objects and events worthy of attention, ignoring things that are not important to current goals and task priorities. It is what enables the world model to answer queries from behavior generating modules, and make predictions and receive updates from sensory processing modules. It is what communicates the value state-variables that describe the success of behavior and the desirability of states of the world from the value judgment element to the goal selection subsystem.

V. A Proposed Architecture For Intelligent Systems

A number of system architectures for intelligent machine systems have been conceived, and a few implemented. [1]–[15] The architecture for intelligent systems that will be proposed here is largely based on the real-time control system (RCS) that has been implemented in a number of versions over the past 13 years at the National Institute for Standards and Technology (NIST, formerly NBS). RCS was first implemented by Barbera for laboratory robotics in the mid 1970's [7] and adapted by Albus, Barbera, and others for manufacturing control in the NIST Automated Manufacturing Research Facility (AMRF) during the early 1980's [11], [12]. Since 1986, RCS has been implemented for a number of additional applications, including the NBS/DARPA Multiple Autonomous Undersea Vehicle (MAUV) project [13], the Army Field Material Handling Robot, and the Army TMAP and TEAM semiautonomous land vehicle projects. RCS also forms the basis of the NASA/NBS Standard Reference Model Telerobot Control System Architecture (NASREM) being used on the space station Flight Telerobotic Servicer [14] and the Air Force Next Generation Controller.

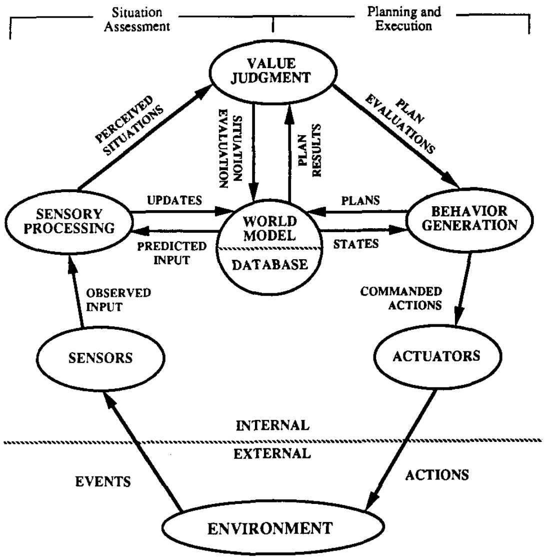


Fig. 1. Elements of intelligence and the functional relationships between them.

The proposed system architecture organizes the elements of intelligence so as to create the functional relationships and information flow shown in Fig. 1. In all intelligent systems, a sensory processing system processes sensory information to acquire and maintain an internal model of the external world. In all systems, a behavior generating system controls actuators so as to pursue behavioral goals in the context of the perceived world model. In systems of higher intelligence, the behavior generating system element may interact with the world model and value judgment system to reason about space and time, geometry and dynamics, and to formulate or select plans based on values such as cost, risk, utility, and goal priorities. The sensory processing system element may interact with the world model and value judgment system to assign values to perceived entities, events, and situations.

The proposed system architecture replicates and distributes the relationships shown in Fig. 1 over a hierarchical computing structure with the logical and temporal properties illustrated in Fig. 2. On the left is an organizational hierarchy wherein computational nodes are arranged in layers like command posts in a military organization. Each node in the organizational hierarchy contains four types of computing modules: behavior generating (BG), world modeling (WM), sensory processing (SP), and value judgment (VJ) modules. Each chain of command in the organizational hierarchy, from each actuator and each sensor to the highest level of control, can be represented by a computational hierarchy, such as is shown in the center of Fig. 2.

At each level, the nodes, and computing modules within the nodes, are richly interconnected to each other by a communications system. Within each computational node, the communication system provides intermodule communications

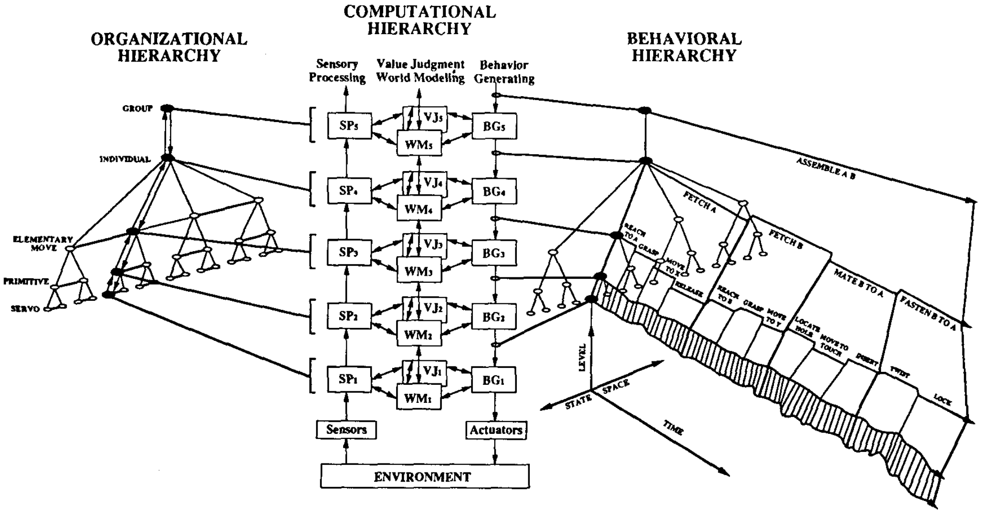


Fig. 2. Relationships in hierarchical control systems, On the left is an organizational hierarchy consisting of a tree of command centers, each of which possesses one supervisor and one or more subordinates. In the center is a computational hierarchy consisting of BG, WM, SP, and VJ modules. Each actuator and each sensors is serviced by a computational hierarchy. On the right is a behavioral hierarchy consisting of trajectories through state–time–space. Commands at a each level can be represented by vectors, or points in state–space. Sequences of commands and be represented as trajectories through state–time–space.

of the type shown in Fig. 1. Queries and task status are communicated from BG modules to WM modules. Retrievals of information are communicated from WM modules back to the BG modules making the queries. Predicted sensory data is communicated from WM modules to SP modules. Updates to the world model are communicated from SP to WM modules. Observed entities, events, and situations are communicated from SP to VJ modules. Values assigned to the world model representations of these entities, events, and situations are communicated from VJ to WM modules. Hypothesized plans are communicated from BG to WM modules. Results are communicated from WM to VJ modules. Evaluations are communicated from VJ modules back to the BG modules that hypothesized the plans.

The communications system also communicates between nodes at different levels. Commands are communicated downward from supervisor BG modules in one level to subordinate BG modules in the level below. Status reports are communicated back upward through the world model from lower level subordinate BG modules to the upper level supervisor BG modules from which commands were received. Observed entities, events, and situations detected by SP modules at one level are communicated upward to SP modules at a higher level. Predicted attributes of entities, events, and situations stored in the WM modules at a higher level are communicated downward to lower level WM modules. Output from the bottom level BG modules is communicated to actuator drive mechanisms. Input to the bottom level SP modules is communicated from sensors.

The communications system can be implemented in a variety of ways. In a biological brain, communication is mostly via neuronal axon pathways, although some messages are communicated by hormones carried in the bloodstream. In artificial systems, the physical implementation of communications functions may be a computer bus, a local area network, a common memory, a message passing system, or some combination thereof. In either biological or artificial systems, the communications system may include the functionality of a communications processor, a file server, a database management system, a question answering system, or an indirect addressing or list processing engine. In the system architecture proposed here, the input/output relationships of the communications system produce the effect of a virtual global memory, or blackboard system [15].

The input command string to each of the BG modules at each level generates a trajectory through state-space as a function of time. The set of all command strings create a behavioral hierarchy, as shown on the right of Fig. 2. Actuator output trajectories (not shown in Fig. 2) correspond to observable output behavior. All the other trajectories in the behavioral hierarchy constitute the deep structure of behavior [16].

VI. Hierarchical Versus Horizontal

Fig. 3 shows the organizational hierarchy in more detail, and illustrates both the hierarchical and horizontal relationships involved in the proposed architecture. The architecture is hierarchical in that commands and status feedback flow hierarchically up and down a behavior generating chain of command. The architecture is also hierarchical in that sensory processing and world modeling functions have hierarchical levels of temporal and spatial aggregation.

The architecture is horizontal in that data is shared horizontally between heterogeneous modules at the same level. At each hierarchical level, the architecture is horizontally interconnected by wide-bandwidth communication pathways between BG, WM, SP, and VJ modules in the same node,

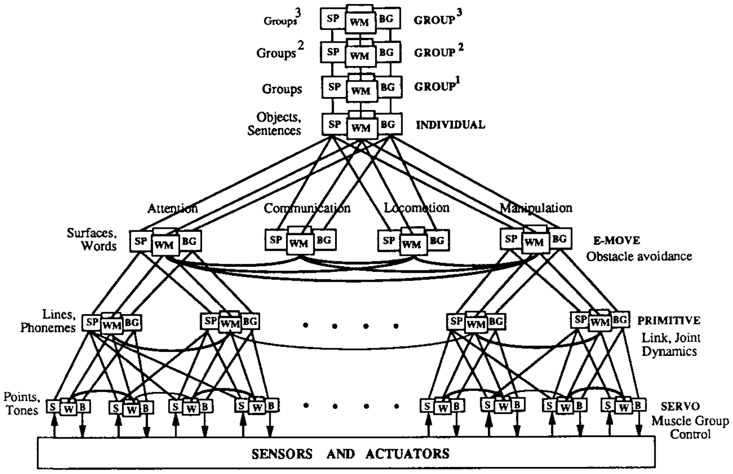


Fig. 3. An organization of processing nodes such that the BG modules form a command tree. On the right are examples or the functional characteristic of the BG modules at each level. On the left are examples of the type of visual and acoustical entities recognized by the SP modules at each level. In the center of level 3 are the type of subsystems represented by processing nodes at level 3.

and between nodes at the same level, especially within the same command subtree. The horizontal flow of information is most voluminous within a single node, less so between related nodes in the same command subtree, and relatively low bandwidth between computing modules in separate command subtrees. Communications bandwidth is indicated in Fig. 3 by the relative thickness of the horizontal connections.

The volume of information flowing horizontally within a subtree may be orders of magnitude larger than the amount flowing vertically in the command chain. The volume of information flowing vertically in the sensory processing system can also be very high, especially in the vision system.

The specific configuration of the command tree is task dependent, and therefore not necessarily stationary in time. Fig. 3 illustrates only one possible configuration that may exist at a single point in time. During operation, relationships between modules within and between layers of the hierarchy may be reconfigured in order to accomplish different goals, priorities, and task requirements. This means that any particular computational node, with its BG, WM, SP, and VJ modules, may belong to one subsystem at one time and a different subsystem a very short time later. For example, the mouth may be part of the manipulation subsystem (while eating) and the communication subsystem (while speaking). Similarly, an arm may be part of the manipulation subsystem (while grasping) and part of the locomotion subsystem (while swimming or climbing).

In the biological brain, command tree reconfiguration can be implemented through multiple axon pathways that exist, but are not always activated, between BG modules at different hierarchical levels. These multiple pathways define a layered graph, or lattice, of nodes and directed arcs, such as shown in Fig. 4. They enable each BG module to receive input messages and parameters from several different sources.

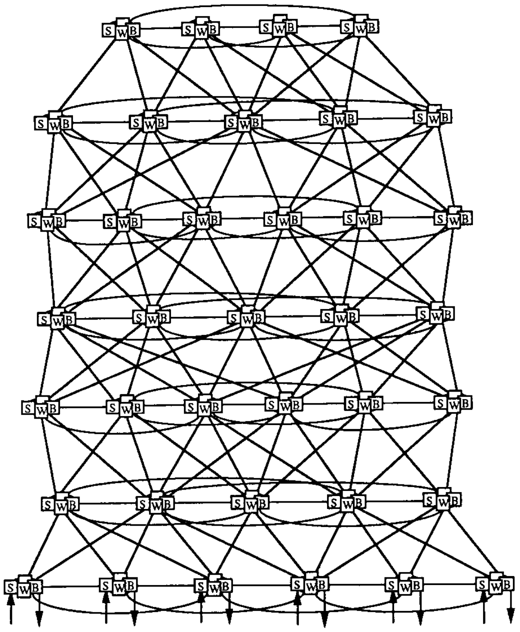


Fig. 4. Each layer of the system architecture contains a number of nodes, each of which contains BG, WM, SP, and VJ modules, The nodes are interconnected as a layered graph, or lattice, through the communication system. Note that the nodes are richly but not fully, interconnected. Outputs from the bottom layer BG modules drive actuators. Inputs to the bottom layer SP modules convey data from sensors. During operation, goal driven communication path selection mechanisms configure this lattice structure into the organization tree shown in Fig. 3.

During operation, goal driven switching mechanisms in the BG modules (discussed in Section X) assess priorities, negotiate for resources, and coordinate task activities so as to select among the possible communication paths of Fig. 4. As a result, each BG module accepts task commands from only one supervisor at a time, and hence the BG modules form a command tree at every instant in time.

The SP modules are also organized hierarchically, but as a layered graph, not a tree. At each higher level, sensory information is processed into increasingly higher levels of abstraction, but the sensory processing pathways may branch and merge in many different ways.

VII. Hierarchical Levels

Levels in the behavior generating hierarchy are defined by temporal and spatial decomposition of goals and tasks into levels of resolution. Temporal resolution is manifested in terms of loop bandwidth, sampling rate, and state-change intervals. Temporal span is measured by the length of historical traces and planning horizons. Spatial resolution is manifested in the branching of the command tree and the resolution of maps. Spatial span is measured by the span of control and the range of maps.

Levels in the sensory processing hierarchy are defined by temporal and spatial integration of sensory data into levels of aggregation. Spatial aggregation is best illustrated by visual images. Temporal aggregation is best illustrated by acoustic parameters such as phase, pitch, phonemes, words, sentences, rhythm, beat, and melody.

Levels in the world model hierarchy are defined by temporal resolution of events, spatial resolution of maps, and by parent-child relationships between entities in symbolic data structures. These are defined by the needs of both SP and BG modules at the various levels.

Theorem: In a hierarchically structured goal-driven, sensory-interactive, intelligent control system architecture:

1) control bandwidth decreases about an order of magnitude at each higher level,
2) perceptual resolution of spatial and temporal patterns decreases about an order-of-magnitude at each higher level,
3) goals expand in scope and planning horizons expand in space and time about an order-of-magnitude at each higher level, and
4) models of the world and memories of events decrease in resolution and expand in spatial and temporal range by about an order-of-magnitude at each higher level.

It is well known from control theory that hierarchically nested servo loops tend to suffer instability unless the bandwidth of the control loops differ by about an order of magnitude. This suggests, perhaps even requires, condition 1). Numerous theoretical and experimental studies support the concept of hierarchical planning and perceptual "chunking" for both temporal and spatial entities [17], [18]. These support conditions 2), 3), and 4).

In elaboration of the aforementioned theorem, we can construct a timing diagram, as shown in Fig. 5. The range of the time scale increases, and its resolution decreases, exponentially by about an order of magnitude at each higher level. Hence the planning horizon and event summary interval increases, and the loop bandwidth and frequency of subgoal events decreases, exponentially at each higher level. The seven hierarchical levels in Fig. 5 span a range of time intervals from three milliseconds to one day. Three milliseconds was arbitrarily chosen as the shortest servo update rate because that is adequate to reproduce the highest bandwidth reflex arc in the human body. One day was arbitrarily chosen as the longest historical-memory/planning-horizon to be considered. Shorter time intervals could be handled by adding another layer at the bottom. Longer time intervals could be treated by adding layers at the top, or by increasing the difference in loop bandwidths and sensory chunking intervals between levels.

The origin of the time axis in Fig. 5 is the present, i.e., $t = 0$. Future plans lie to the right of $t = 0$, past history to the left. The open triangles in the right half-plane represent task goals in a future plan. The filled triangles in the left half-plane represent recognized task-completion events in a past history. At each level there is a planning horizon and a historical event summary interval. The heavy crosshatching on the right shows the planning horizon for the current task. The light shading on the right indicates the planning horizon for the anticipated next task. The heavy crosshatching on the left shows the event summary interval for the current task. The light shading on the left shows the event summary interval for the immediately previous task.

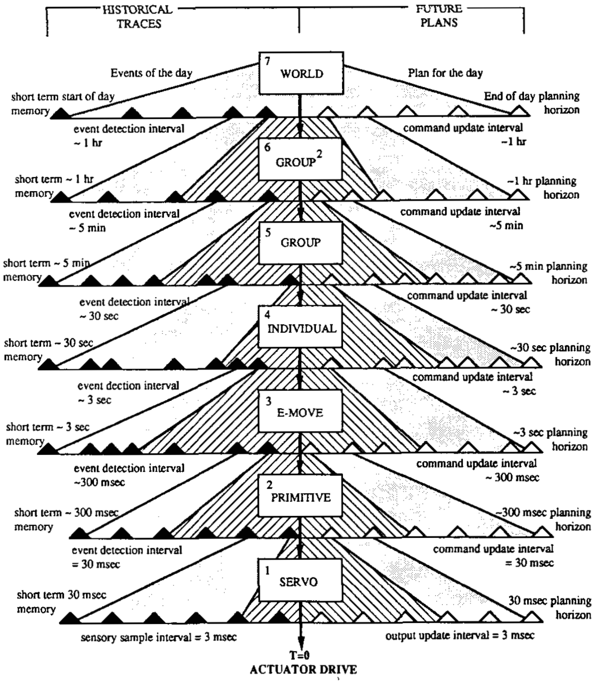


Fig. 5. Timing diagram illustrating the temporal flow of activity in the task decomposition and sensory processing systems. At the world level, high-level sensory events and circadian rhythms react with habits and daily routines to generate a plan for the day. Each elements of that plan is decomposed through the remaining six levels of task decomposition into action.

Fig. 5 suggests a duality between the behavior generation and the sensory processing hierarchies. At each hierarchical level, planner modules decompose task commands into strings of planned subtasks for execution. At each level, strings of sensed events are summarized, integrated, and "chunked" into single events at the next higher level.

Planning implies an ability to predict future states of the world. Prediction algorithms based on Fourier transforms or Kalman filters typically use recent historical data to compute parameters for extrapolating into the future. Predictions made by such methods are typically not reliable for periods longer than the historical interval over which the parameters were computed. Thus at each level, planning horizons extend into the future only about as far, and with about the same level of detail, as historical traces reach into the past.

Predicting the future state of the world often depends on assumptions as to what actions are going to be taken and what reactions are to be expected from the environment, including what actions may be taken by other intelligent agents. Planning of this type requires search over the space of possible future actions and probable reactions. Search-based planning takes place via a looping interaction between the BG, WM, and VJ modules. This is described in more detail in the Section X discussion on BG modules.

Planning complexity grows exponentially with the number

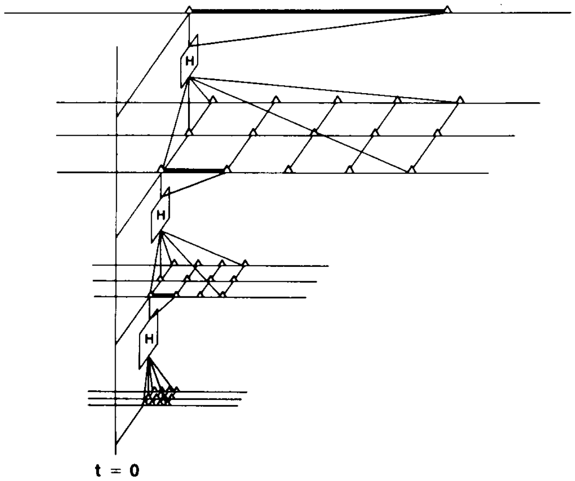


Fig. 6. Three levels of real-time planning illustrating the shrinking planning horizon and greater detail at successively lower levels of the hierarchy. At the top level, a single task is decomposed into a set of four planned subtasks for each of three subsystem. At each of the next two levels, the first task in the plan of the first subsystems is further decomposed into four subtasks for three subsystems at the next lower level.

of steps in the plan (i.e., the number of layers in the search graph). If real-time planning is to succeed, any given planner must operate in a limited search space. If there are too much resolution in the time line, or in the space of possible actions, the size of the search graph can easily become too large for real-time response. One method of resolving this problem is to use a multiplicity of planners in hierarchical layers [14], [18] so that at each layer no planner needs to search more than a given number (for example ten) steps deep in a game graph, and at each level there are no more than (ten) subsystem planners that need to simultaneously generate and coordinate plans. These criteria give rise to hierarchical levels with exponentially expanding spatial and temporal planning horizons, and characteristic degrees of detail for each level. The result of hierarchical spatiotemporal planning is illustrated in Fig. 6. At each level, plans consist of at least one, and on average 10, subtasks. The planners have a planning horizon that extends about one and a half average input command intervals into the future.

In a real-time system, plans must be regenerated periodically to cope with changing and unforeseen conditions in the world. Cyclic replanning may occur at periodic intervals. Emergency replanning begins immediately upon the detection of an emergency condition. Under full alert status, the cyclic replanning interval should be about an order of magnitude less than the planning horizon (or about equal to the expected output subtask time duration). This requires that real-time planners be able to search to the planning horizon about an order of magnitude faster than real time. This is possible only if the depth and resolution of search is limited through hierarchical planning.

Plan executors at each level have responsibility for reacting to feedback every control cycle interval. Control cycle intervals are inversely proportional to the control loop bandwidth. Typically the control cycle interval is an order of magnitude less than the expected output subtask duration. If the feedback indicates the failure of a planned subtask, the executor branches immediately (i.e., in one control cycle interval) to a preplanned emergency subtask. The planner simultaneously selects or generates an error recovery sequence that is substituted for the former plan that failed. Plan executors are also described in more detail in Section X.

When a task goal is achieved at time $t = 0$, it becomes a task completion event in the historical trace. To the extent that a historical trace is an exact duplicate of a former plan, there were no surprises; i.e., the plan was followed, and every task was accomplished as planned. To the extent that a historical trace is different from the former plan, there were surprises. The average size and frequency of surprises (i.e., differences between plans and results) is a measure of effectiveness of a planner.

At each level in the control hierarchy, the difference vector between planned (i.e., predicted) commands and observed events is an error signal, that can be used by executor submodules for servo feedback control (i.e., error correction), and by VJ modules for evaluating success and failure.

In the next eight sections, the system architecture outlined previously will be elaborated and the functionality of the computational submodules for behavior generation, world modeling, sensory processing, and value judgment will be discussed.

VIII. Behavior Generation

Definition: Behavior is the result of executing a series of tasks.

Definition: A task is a piece of work to be done, or an activity to be performed.

Axiom: For any intelligent system, there exists a set of tasks that the system knows how to do.

Each task in this set can be assigned a name. The task vocabulary is the set of task names assigned to the set of tasks the system is capable of performing. For creatures capable of learning, the task vocabulary is not fixed in size. It can be expanded through learning, training, or programming. It may shrink from forgetting, or program deletion.

Typically, a task is performed by a one or more actors on one or more objects. The performance of a task can usually be described as an activity that begins with a start-event and is directed toward a goal-event. This is illustrated in Fig. 7.

Definition: A goal is an event that successfully terminates a task. A goal is the objective toward which task activity is directed.

Definition: A task command is an instruction to perform a named task. A task command may have the form: DO <Taskname(parameters)> AFTER <Start Event> UNTIL <Goal Event> Task knowledge is knowledge of how to perform a task, including information as to what tools, materials, time, resources, information, and conditions are required, plus information as to what costs, benefits and risks are expected.

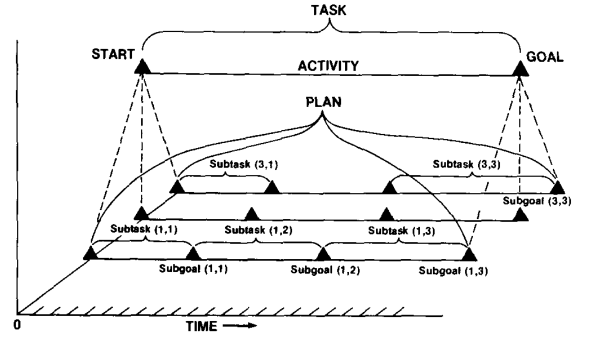


Fig. 7. A task consists of an activity that typically begins with a start event and is terminated by a goal event. A task may be decomposed into several concurrent strings of subtasks that collectively achieve the goal event.

Task knowledge may be expressed implicitly in fixed circuitry, either in the neuronal connections and synaptic weights of the brain, or in algorithms, software, and computing hardware. Task knowledge may also be expressed explicitly in data structures, either in the neuronal substrate or in a computer memory.

Definition: A task frame is a data structure in which task knowledge can be stored.

In systems where task knowledge is explicit, a task frame [19] can be defined for each task in the task vocabulary. An example of a task frame is:

TASKNAME	name of the task
type	generic or specifi
actor	agent performing the task
action	activity to be performed
object	thing to be acted upon
goal	event that successfully terminates or renders the task successful
parameters	priority status (e.g. active, waiting, inactive) timing requirements source of task command
requirements	tools, time, resources, and materials needed to perform the task enabling conditions that must be satisfied to begin, or continue, the task disabling conditions that will prevent, or interrupt, the task information that may be required
procedures	a state-graph or state-table defining a plan for executing the task functions that may be called algorithms that may be needed
effects	expected results of task execution expected costs, risks, benefits estimated time to complete

Explicit representation of task knowledge in task frames has a variety of uses. For example, task planners may use it for generating hypothesized actions. The world model may use it for predicting the results of hypothesized actions. The value judgment system may use it for computing how important the goal is and how many resources to expend in pursuing it. Plan executors may use it for selecting what to do next.

Task knowledge is typically difficult to discover, but once known, can be readily transferred to others. Task knowledge may be acquired by trial and error learning, but more often it is acquired from a teacher, or from written or programmed instructions. For example, the common household task of preparing a food dish is typically performed by following a recipe. A recipe is an informal task frame for cooking. Gourmet dishes rarely result from reasoning about possible combinations of ingredients, still less from random trial and error combinations of food stuffs. Exceptionally good recipes often are closely guarded secrets that, once published, can easily be understood and followed by others.

Making steel is a more complex task example. Steel making took the human race many millennia to discover how to do. However, once known, the recipe for making steel can be implemented by persons of ordinary skill and intelligence.

In most cases, the ability to successfully accomplish complex tasks is more dependent on the amount of task knowledge stored in task frames (particularly in the procedure section) than on the sophistication of planners in reasoning about tasks.

IX. Behavior Generation

Behavior generation is inherently a hierarchical process. At each level of the behavior generation hierarchy, tasks are decomposed into subtasks that become task commands to the next lower level. At each level of a behavior generation hierarchy there exists a task vocabulary and a corresponding set of task frames. Each task frame contains a procedure state-graph. Each node in the procedure state-graph must correspond to a task name in the task vocabulary at the next lower level.

Behavior generation consists of both spatial and temporal decomposition. Spatial decomposition partitions a task into jobs to be performed by different subsystems. Spatial task decomposition results in a tree structure, where each node corresponds to a BG module, and each arc of the tree corresponds to a communication link in the chain of command as illustrated in Fig. 3.

Temporal decomposition partitions each job into sequential subtasks along the time line. The result is a set of subtasks, all of which when accomplished, achieve the task goal, as illustrated in Fig. 7.

In a plan involving concurrent job activity by different subsystems, there may requirements for coordination, or mutual constraints. For example, a start-event for a subtask activity in one subsystem may depend on the goal-event for a subtask activity in another subsystem. Some tasks may require concurrent coordinated cooperative action by several subsystems. Both planning and execution of subsystem plans may thus need to be coordinated.

There may be several alternative ways to accomplish a task. Alternative task or job decompositions can be represented by an AND/OR graph in the procedure section of the task frame. The decision as to which of several alternatives to choose is made through a series of interactions between the BG, WM, SP, and VJ modules. Each alternative may be analyzed by the BG module hypothesizing it, WM predicting the result, and VJ

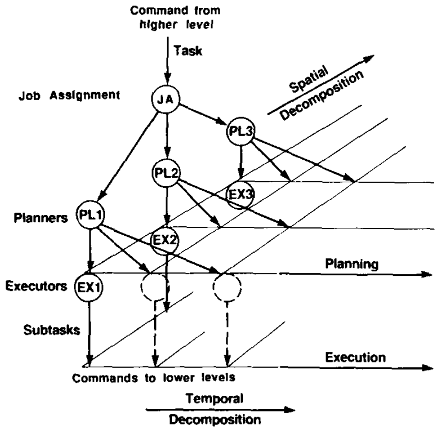


Fig. 8. The job assignment JA module performs a spatial decomposition of the task command into N subsystems. For each subsystem, a planner $PL(j)$ performs a temporal decomposition of its assigned job into subtasks. For each subsystem, an executor $EX(j)$ closes a real-time control loop that servos the subtasks to the plan.

evaluating the result. The BG module then chooses the "best" alternative as the plan to be executed.

X. BG Modules

In the control architecture defined in Fig. 3, each level of the hierarchy contains one or more BG modules. At each level, there is a BG module for each subsystem being controlled. The function of the BG modules are to decompose task commands into subtask commands.

Input to BG modules consists of commands and priorities from BG modules at the next higher level, plus evaluations from nearby VJ modules, plus information about past, present, and predicted future states of the world from nearby WM modules. Output from BG modules may consist of subtask commands to BG modules at the next lower level, plus status reports, plus "What Is?" and "What If?" queries to the WM about the current and future states of the world.

Each BG module at each level consists of three sublevels [9], [14] as shown in Fig. 8.

The Job Assignment Sublevel—JA Submodule: The JA submodule is responsible for spatial task decomposition. It partitions the input task command into N spatially distinct jobs to be performed by N physically distinct subsystems, where N is the number of subsystems currently assigned to the BG module. The JA submodule many assign tools and allocate physical resources (such as arms, hands, legs, sensors, tools, and materials) to each of its subordinate subsystems for their use in performing their assigned jobs. These assignments are not necessarily static. For example, the job assignment submodule at the individual level may, at one moment, assign an arm to the manipulation subsystem in response to a <usetool> task command, and later, assign the same arm to the attention subsystem in response to a <touch/feel> task command.

The job assignment submodule selects the coordinate system in which the task decomposition at that level is to be performed. In supervisory or telerobotic control systems such as defined by NASREM [14], the JA submodule at each level may also determine the amount and kind of input to accept from a human operator.

The Planner Sublevel—PL(j) Submodules j=1, 2, . . .N: For each of the N subsystems, there exists a planner submodule $PL(j)$. Each planner submodule is responsible for decomposing the job assigned to its subsystem into a temporal sequence of planned subtasks.

Planner submodules $PL(j)$ may be implemented by case-based planners that simply select partially or completely prefabricated plans, scripts, or schema [20]–[22] from the procedure sections of task frames. This may be done by evoking situation/action rules of the form, IF(case_x)/THEN(use_plan_y). The planner submodules may complete partial plans by providing situation dependent parameters.

The range of behavior that can be generated by a library of prefabricated plans at each hierarchical level, with each plan containing a number of conditional branches and error recovery routines, can be extremely large and complex. For example, nature has provided biological creatures with an extensive library of genetically prefabricated plans, called instinct. For most species, case-based planning using libraries of instinctive plans has proven adequate for survival and gene propagation in a hostile natural environment.

Planner submodules may also be implemented by search-based planners that search the space of possible actions. This requires the evaluation of alternative hypothetical sequences of subtasks, as illustrated in Fig. 9. Each planner $PL(j)$ hypothesizes some action or series of actions, the WM module predicts the effects of those action(s), and the VJ module computes the value of the resulting expected states of the world, as depicted in Fig. 9(a). This results in a game (or search) graph, as shown in 9(b). The path through the game graph leading to the state with the best value becomes the plan to be executed by $EX(j)$. In either case-based or search-based planning, the resulting plan may be represented by a state-graph, as shown in Fig. 9(c). Plans may also be represented by gradients, or other types of fields, on maps [23], or in configuration space.

Job commands to each planner submodule may contain constraints on time, or specify job-start and job-goal events. A job assigned to one subsystem may also require synchronization or coordination with other jobs assigned to different subsystems. These constraints and coordination requirements may be specified by, or derived from, the task frame. Each planner $PL(j)$ submodule is responsible for coordinating its plan with plans generated by each of the other $N - 1$ planners at the same level, and checking to determine if there are mutually conflicting constraints. If conflicts are found, constraint relaxation algorithms [24] may be applied, or negotiations conducted between $PL(j)$ planners, until a solution is discovered. If no solution can be found, the planners report failure to the job assignment submodule, and a new job assignment may be tried, or failure may be reported to the next higher level BG module.

The Executor Sublevel—EX(j) Submodules: There is an executor $EX(j)$ for each planner $PL(j)$. The executor submodules are responsible for successfully executing the plan

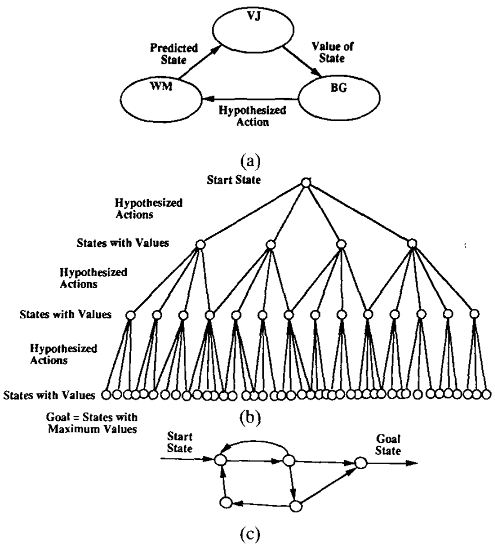


Fig. 9. Planning loop (a) produces a game graph (b). A trace in the game graph from the start to a goal state is a plan that can be represented as a plan graph (c). Nodes in the game graph correspond to edges in the plan graph, and edges in the game graph correspond to nodes in the plan graph. Multiple edges exiting nodes in the plan graph correspond to conditional branches.

state-graphs generated by their respective planners. At each tick of the state clock, each executor measures the difference between the current world state and its current plan subgoal state, and issues a subcommand designed to null the difference. When the world model indicates that a subtask in the current plan is successfully completed, the executor steps to the next subtask in that plan. When all the subtasks in the current plan are successfully executed, the executor steps to the first subtask in the next plan. If the feedback indicates the failure of a planned subtask, the executor branches immediately to a preplanned emergency subtask. Its planner meanwhile begins work selecting or generating a new plan that can be substituted for the former plan that failed. Output subcommands produced by executors at level i become input commands to job assignment submodules in BG modules at level $i - 1$.

Planners $PL(j)$ operate on the future. For each subsystem, there is a planner that is responsible for providing a plan that extends to the end of its planning horizon. Executors $EX(j)$ operate in the present. For each subsystem, there is an executor that is responsible for monitoring the current ($t = 0$) state of the world and executing the plan for its respective subsystem. Each executor performs a READ-COMPUTE-WRITE operation once each control cycle. At each level, each executor submodule closes a reflex arc, or servo loop. Thus, executor submodules at the various hierarchical levels form a set of nested servo loops. Executor loop bandwidths decrease on average about an order of magnitude at each higher level.

XI. The Behavior Generating Hierarchy

Task goals and task decomposition functions often have characteristic spatial and temporal properties. For any task, there exists a hierarchy of task vocabularies that can be overlaid on the spatial/temporal hierarchy of Fig. 5.

For example:

Level 1 is where commands for coordinated velocities and forces of body components (such as arms, hands, fingers, legs, eyes, torso, and head) are decomposed into motor commands to individual actuators. Feedback servos the position, velocity, and force of individual actuators. In vertebrates, this is the level of the motor neuron and stretch reflex.

Level 2 is where commands for maneuvers of body components are decomposed into smooth coordinated dynamically efficient trajectories. Feedback servos coordinated trajectory motions. This is the level of the spinal motor centers and the cerebellum.

Level 3 is where commands to manipulation, locomotion, and attention subsystems are decomposed into collision free paths that avoid obstacles and singularities. Feedback servos movements relative to surfaces in the world. This is the level of the red nucleus, the substantia nigra, and the primary motor cortex.

Level 4 is where commands for an individual to perform simple tasks on single objects are decomposed into coordinated activity of body locomotion, manipulation, attention, and communication subsystems. Feedback initiates and sequences subsystem activity. This is the level of the basal ganglia and pre-motor frontal cortex.

Level 5 is where commands for behavior of an intelligent self individual relative to others in a small group are decomposed into interactions between the self and nearby objects or agents. Feedback initiates and steers whole self task activity. Behavior generating levels 5 and above are hypothesized to reside in temporal, frontal, and limbic cortical areas.

Level 6 is where commands for behavior of the individual relative to multiple groups are decomposed into small group interactions. Feedback steers small group interactions.

Level 7 (arbitrarily the highest level) is where long range goals are selected and plans are made for long range behavior relative to the world as a whole. Feedback steers progress toward long range goals.

The mapping of BG functionality onto levels one to four defines the control functions necessary to control a single intelligent individual in performing simple task goals. Functionality at levels one through three is more or less fixed and specific to each species of intelligent system [25]. At level 4 and above, the mapping becomes more task and situation dependent. Levels 5 and above define the control functions necessary to control the relationships of an individual relative to others in groups, multiple groups, and the world as a whole.

There is good evidence that hierarchical layers develop in the sensory-motor system, both in the individual brain as the individual matures, and in the brains of an entire species as the species evolves. It can be hypothesized that the maturation of levels in humans gives rise to Piaget's "stages of development" [26].

Of course, the biological motor system is typically much more complex than is suggested by the example model described previously. In the brains of higher species there may exist multiple hierarchies that overlap and interact with each other in complicated ways. For example in primates, the pyramidal cells of the primary motor cortex have outputs to the motor neurons for direct control of fine manipulation as well as the inferior olive for teaching behavioral skills to the cerebellum [27]. There is also evidence for three parallel behavior generating hierarchies that have developed over three evolutionary eras [28]. Each BG module may thus contain three or more competing influences: 1) the most basic (IF it smells good, THEN eat it), 2) a more sophisticated (WAIT until the "best" moment) where best is when success probability is highest, and 3) a very sophisticated (WHAT are the long range consequences of my contemplated action, and what are all my options).

On the other hand, some motor systems may be less complex than suggested previously. Not all species have the same number of levels. Insects, for example, may have only two or three levels, while adult humans may have more than seven. In robots, the functionality required of each BG module depends upon the complexity of the subsystem being controlled. For example, one robot gripper may consist of a dexterous hand with 15 to 20 force servoed degrees of freedom. Another gripper may consist of two parallel jaws actuated by a single pneumatic cylinder. In simple systems, some BG modules (such as the Primitive level) may have no function (such as dynamic trajectory computation) to perform. In this case, the BG module will simply pass through unchanged input commands (such as <Grasp>).

XII. The World Model

Definition: The world model is an intelligent system's internal representation of the external world. It is the system's best estimate of objective reality. A clear distinction between an internal representation of the world that exists in the mind, and the external world of reality, was first made in the West by Schopenhauer over 100 years ago [29]. In the East, it has been a central theme of Buddhism for millennia. Today the concept of an internal world model is crucial to an understanding of perception and cognition. The world model provides the intelligent system with the information necessary to reason about objects, space, and time. The world model contains knowledge of things that are not directly and immediately observable. It enables the system to integrate noisy and intermittent sensory input from many different sources into a single reliable representation of spatiotemporal reality.

Knowledge in an intelligent system may be represented either implicitly or explicitly. Implicit world knowledge may be embedded in the control and sensory processing algorithms and interconnections of a brain, or of a computer system. Explicit world knowledge may be represented in either natural or artificial systems by data in database structures such as maps, lists, and semantic nets. Explicit world models require computational modules capable of map transformations, indirect addressing, and list processing. Computer hardware and software techniques for implementing these types of functions are well known. Neural mechanisms with such capabilities are discussed in Section XVI.

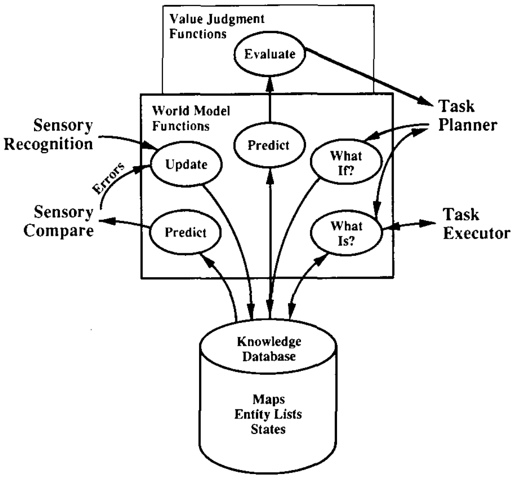


Fig. 10. Functions performed by the WM module. 1) Update knowledge database with prediction errors and recognized entities. 2) Predict sensory data. 3) Answer "What is?" queries from task executor and return current state of world. 4) Answer "What if?" queries from task planner and predict results for evaluation.

A. WM Modules

The WM modules in each node of the organizational hierarchy of Figs. 2 and 3 perform the functions illustrated in Fig. 10.

1) WM modules maintain the knowledge database, keeping it current and consistent. In this role, the WM modules perform the functions of a database management system. They update WM state estimates based on correlations and differences between world model predictions and sensory observations at each hierarchical level. The WM modules enter newly recognized entities, states, and events into the knowledge database, and delete entities and states determined by the sensory processing modules to no longer exist in the external world. The WM modules also enter estimates, generated by the VJ modules, of the reliability of world model state variables. Believability or confidence factors are assigned to many types of state variables.
2) WM modules generate predictions of expected sensory input for use by the appropriate sensory processing SP modules. In this role, a WM module performs the functions of a signal generator, a graphics engine, or state predictor, generating predictions that enable the sensory processing system to perform correlation and predictive filtering. WM predictions are based on the state of the task and estimated states of the external world. For example in vision, a WM module may use the information in an object frame to generate real-time predicted images that can be compared pixel by pixel, or entity by entity, with observed images.
3) WM modules answer "What is?" questions asked by the planners and executors in the corresponding level BG modules. In this role, the WM modules perform the function of database query processors, question answering

systems, or data servers. World model estimates of the current state of the world are also used by BG module planners as a starting point for planning. Current state estimates are used by BG module executors for servoing and branching on conditions.

4) WM modules answer "What if?" questions asked by the planners in the corresponding level BG modules. In this role, the WM modules perform the function of simulation by generating expected status resulting from actions hypothesized by the BG planners. Results predicted by WM simulations are sent to value judgment VJ modules for evaluation. For each BG hypothesized action, a WM prediction is generated, and a VJ evaluation is returned to the BG planner. This BG-WM-VJ loop enables BG planners to select the sequence of hypothesized actions producing the best evaluation as the plan to be executed.

Data structures for representing explicit knowledge are defined to reside in a knowledge database that is hierarchically structured and distributed such that there is a knowledge database for each WM module in each node at every level of the system hierarchy. The communication system provides data transmission and switching services that make the WM modules and the knowledge database behave like a global virtual common memory in response to queries and updates from the BG, SP, and VJ modules. The communication interfaces with the WM modules in each node provides a window into the knowledge database for each of the computing modules in that node.

XIII. Knowledge Representation

The world model knowledge database contains both *a priori* information that is available to the intelligent system before action begins, and *a posteriori* knowledge that is gained from sensing the environment as action proceeds. It contains information about space, time, entities, events, and states of the external world. The knowledge database also includes information about the intelligent system itself, such as values assigned to motives, drives, and priorities; values assigned to goals, objects, and events; parameters embedded in kinematic and dynamic models of the limbs and body; states of internal pressure, temperature, clocks, and blood chemistry or fuel level; plus the states of all of the processes currently executing in each of the BG, SP, WM, and VJ modules.

Knowledge about space is represented in maps. Knowledge about entities, events, and states is represented in lists, or frames. Knowledge about the laws of physics, chemistry, optics, and the rules of logic and mathematics are represented as parameters in the WM functions that generate predictions and simulate results of hypothetical actions. Physical knowledge may be represented as algorithms, formulae, or as IF/THEN rules of what happens under certain situations, such as when things are pushed, thrown, dropped, handled, or burned.

The correctness and consistency of world model knowledge is verified by sensory processing mechanisms that measure differences between world model predictions and sensory observations.

A. Geometrical Space

From psychophysical evidence Gibson [30] concludes that the perception of geometrical space is primarily in terms of "medium, substance, and the surfaces that separate them". Medium is the air, water, fog, smoke, or falling snow through which the world is viewed. Substance is the material, such as earth, rock, wood, metal, flesh, grass, clouds, or water, that comprise the interior of objects. The surfaces that separate the viewing medium from the viewed objects is what are observed by the sensory system. The sensory input thus describes the external physical world primarily in terms of surfaces.

Surfaces are thus selected as the fundamental element for representing space in the proposed WM knowledge database. Volumes are treated as regions between surfaces. Objects are defined as circumscribed, often closed, surfaces. Lines, points and vertices lie on, and may define surfaces. Spatial relationships on surfaces are represented by maps.

B. Maps

Definition: A map is a two dimensional database that defines a mesh or grid on a surface.

The surface represented by a map may be, but need not be, flat. For example, a map may be defined on a surface that is draped over, or even wrapped around, a three-dimensional (3–D) volume.

Theorem: Maps can be used to describe the distribution of entities in space.

It is always possible and often useful to project the physical 3-D world onto a 2-D surface defined by a map. For example, most commonly used maps are produced by projecting the world onto the 2-D surface of a flat sheet of paper, or the surface of a globe. One great advantage of such a projection is that it reduces the dimensionality of the world from three to two. This produces an enormous saving in the amount of memory required for a database representing space. The saving may be as much as three orders of magnitude, or more, depending on the resolution along the projected dimension.

1) Map Overlays: Most of the useful information lost in the projection from 3-D space to a 2-D surface can be recovered through the use of map overlays.

Definition: A map overlay is an assignment of values, or parameters, to points on the map.

A map overlay can represent spatial relationships between 3-D objects. For example, an object overlay may indicate the presence of buildings, roads, bridges, and landmarks at various places on the map. Objects that appear smaller than a pixel on a map can be represented as icons. Larger objects may be represented by labeled regions that are projections of the 3-D objects on the 2-D map. Objects appearing on the map overlay may be cross referenced to an object frame database elsewhere in the world model. Information about the 3-D geometry of objects on the map may be represented in the object frame database.

Map overlays can also indicate attributes associated with points (or pixels) on the map. One of the most common map overlays defines terrain elevation. A value of terrain elevation

(z) overlaid at each (x, y) point on a world map produces a topographic map.

A map can have any number of overlays. Map overlays may indicate brightness, color, temperature, even "behind" or "in-front". A brightness or color overlay may correspond to a visual image. For example, when aerial photos or satellite images are registered with map coordinates, they become brightness or color map overlays.

Map overlays may indicate terrain type, or region names, or can indicate values, such as cost or risk, associated with regions. Map overlays can indicate which points on the ground are visible from a given location in space. Overlays may also indicate contour lines and grid lines such as latitude and longitude, or range and bearing.

Map overlays may be useful for a variety of functions. For example, terrain elevation and other characteristics may be useful for route planning in tasks of manipulation and locomotion. Object overlays can be useful for analyzing scenes and recognizing objects and places.

A map typically represents the configuration of the world at a single instant in time, i.e., a snapshot. Motion can be represented by overlays of state variables such as velocity or image flow vectors, or traces (i.e., trajectories) of entity locations. Time may be represented explicitly by a numerical parameter associated with each trajectory point, or implicitly by causing trajectory points to fade, or be deleted, as time passes.

Definition: A map pixel frame is a frame that contains attributes and attribute-values attached to that map pixel.

Theorem: A set of map overlays are equivalent to a set of map pixel frames.

Proof: If each map overlay defines a parameter value for every map pixel, then the set of all overlay parameter values for each map pixel defines a frame for that pixel. Conversely, the frame for each pixel describes the region covered by that pixel. The set of all pixel frames thus defines a set of map overlays, one overlay for each attribute in the pixel frames.

Q.E.D.

For example, a pixel frame may describe the color, range, and orientation of the surface covered by the pixel. It may describe the name of (or pointer to) the entities to which the surface covered by the pixel belongs. It may also contain the location, or address, of the region covered by the pixel in other coordinate systems.

In the case of a video image, a map pixel frame might have the following form:

PIXEL_NAME	$(AZ.EL)$ location index on map (Sensor egosphere coordinates)
brightness	I
color	$I_r.I_b.I_g$
spatial brightness gradient	$dI/dAZ.\ dI/dEL$ (sensor egosphere coordinates)
temporal brightness gradient	dI/dt
image flow direction	B (velocity egosphere coordinates)
image flow rate	dA/dt (velocity egosphere coordinates)
range	R to surface covered (from egosphere origin)
head egosphere location	az, el of egosphere ray to surface covered
world map location	$x.y.z$ of map point on surface covered
world map location	$x.y.z$ of map point on surface covered
linear feature pointer	pointer to frame of line, edge, or vertex covered by pixel
surface feature pointer	pointer to frame of surface covered by pixel
object pointer	*pointer to frame of object covered* by pixel
object map location	$X.Y.Z$ of surface covered in object coordinates group pointer pointer to group covered by pixel

Indirect addressing through pixel frame pointers can allow value state-variables assigned to objects or situations to be inherited by map pixels. For example, value state-variables such as attraction-repulsion, love-hate, fear-comfort assigned to objects and map regions can also be assigned through inheritance to individual map and egosphere pixels.

There is some experimental evidence to suggest that map pixel frames exist in the mammalian visual system. For example, neuron firing rates in visual cortex have been observed to represent the values of attributes such as edge orientation, edge and vertex type, and motion parameters such as velocity, rotation, and flow field divergence. These firing rates are observed to be registered with retinotopic brightness images [31], [54].

C. Map Resolution

The resolution required for a world model map depends on how the map is generated and how it is used. All overlays need not have the same resolution. For predicting sensory input, world model maps should have resolution comparable to the resolution of the sensory system. For vision, map resolution may be on the order of 64K to a million pixels. This corresponds to image arrays of 256×256 pixels to 1000×1000 pixels respectively. For other sensory modalities, resolution can be considerably less.

For planning, different levels of the control hierarchy require maps of different scale. At higher levels, plans cover long distances and times, and require maps of large area, but low resolution. At lower levels, plans cover short distances and times, and maps need to cover small areas with high resolution. [18]

World model maps generated solely from symbolic data in long term memory may have resolution on the order of a few thousand pixels or less. For example, few humans can recall from memory the relative spatial distribution of as many as a hundred objects, even in familiar locations such as their own homes. The long term spatial memory of an intelligent creature typically consists of a finite number of relatively small regions that may be widely separated in space; for example, one's own home, the office, or school, the homes of friends and relatives, etc. These known regions are typically connected by linear pathways that contain at most a few hundred known waypoints and branchpoints. The remainder of the world is known little, or not at all. Unknown regions, which make up the vast majority of the real world, occupy little or no space in the world model.

The efficient storage of maps with extremely nonuniform resolution can be accomplished in a computer database by quadtrees [32], hash coding, or other sparse memory representations [33]. Pathways between known areas can be economically represented by graph structures either in neuronal or electronic memories. Neural net input-space representations and transformations such as are embodied in a CMAC [34], [35] give insight as to how nonuniformly dense spatial information might be represented in the brain.

D. Maps and Egospheres

It is well known that neurons in the brain, particularly in the cortex, are organized as 2-D arrays, or maps. It is also known that conformal mappings of image arrays exist between the retina, the lateral geniculate, the superior colliculus, and several cortical visual areas. Similar mappings exist in the auditory and tactile sensory systems. For every map, there exists a coordinate system, and each map pixel has coordinate values. On the sensor egosphere, pixel coordinates are defined by the physical position of the pixel in the sensor array. The position of each pixel in other map coordinate systems can be defined either by neuronal interconnections, or by transform parameters contained in each pixel's frame.

There are three general types of map coordinate systems that are important to an intelligent system: world coordinates, object coordinates, and egospheres.

1) World Coordinates: World coordinate maps are typically flat 2-D arrays that are projections of the surface of the earth along the local vertical. World coordinates are often expressed in a Cartesian frame, and referenced to a point in the world. In most cases, the origin is an arbitrary point on the ground. The z axis is defined by the vertical, and the x and y axes define points on the horizon. For example, y may point North and x East. The value of z is often set to zero at sea level.

World coordinates may also be referenced to a moving point in the world. For example, the origin may be the self, or some moving object in the world. In this case, stationary pixels on the world map must be scrolled as the reference point moves.

There may be several world maps with different resolutions and ranges. These will be discussed near the end of this section.

2) Object Coordinates: Object coordinates are defined with respect to features in an object. For example, the origin might be defined as the center of gravity, with the coordinate axes defined by axes of symmetry, faces, edges, vertices, or skeletons [36]. There are a variety of surface representations that have been suggested for representing object geometry. Among these are generalized cylinders [37], [38], B-splines [39], quadtrees [32], and aspect graphs [40]. Object coordinate maps are typically 2-D arrays of points painted on the surfaces of objects in the form of a grid or mesh. Other boundary representation can usually be transformed into this form.

Object map overlays can indicate surface characteristics such as texture, color, hardness, temperature, and type of material. Overlays can be provided for edges, boundaries, surface normal vectors, vertices, and pointers to object frames containing center lines, centroids, moments, and axes of symmetry.

3) Egospheres: An egosphere is a two-dimensional (2-D) spherical surface that is a map of the world as seen by an observer at the center of the sphere. Visible points on regions or objects in the world are projected on the egosphere wherever the line of sight from a sensor at the center of the egosphere to the points in the world intersect the surface of the sphere. Egosphere coordinates thus are polar coordinates defined by the self at the origin. As the self moves, the projection of the world flows across the surface of the egosphere.

Just as the world map is a flat 2-D (x, y) array with multiple overlays, so the egosphere is a spherical 2-D (AZ, EL) array with multiple overlays. Egosphere overlays can attribute brightness, color, range, image flow, texture, and other properties to regions and entities on the egosphere. Regions on the egosphere can thus be segmented by attributes, and egosphere points with the same attribute value may be connected by contour lines. Egosphere overlays may also indicate the trace, or history, of brightness values or entity positions over some time interval. Objects may be represented on the egosphere by icons, and each object may have in its database frame a trace, or trajectory, of positions on the egosphere over some time interval.

E. Map Transformations

Theorem: If surfaces in real world space can be covered by an array (or map) of points in a coordinate system defined in the world, and the surface of a WM egosphere is also represented as an array of points, then there exists a function G that transforms each point on the real world map into a point on the WM egosphere, and a function G' that transforms each point on the WM egosphere for which range is known into a point on the real world map.

Proof: Fig. 11 shows the 3-D relationship between an egosphere and world map coordinates. For every point (x, y, z) in world coordinates, there is a point (AZ, EL, R) in ego centered coordinates that can be computed by the 3×3 matrix function G

$$(AZ, EL, R)^T = G(x, y, z)^T$$

There, of course, may be more than one point in the world map that gives the same (AZ, EL) values on the egosphere. Only the (AZ, EL) with the smallest value of R will be visible to an observer at the center of the egosphere. The deletion of egosphere pixels with R larger than the smallest for each value of (AZ, EL) corresponds to the hidden surface removal problem common ih computer graphics.

For each egosphere pixel where R is known, (x, y, z) can be computed from (AZ, EL, R) by the function G'

$$(x, y, z)^T = G'(AZ, EL, R)^T$$

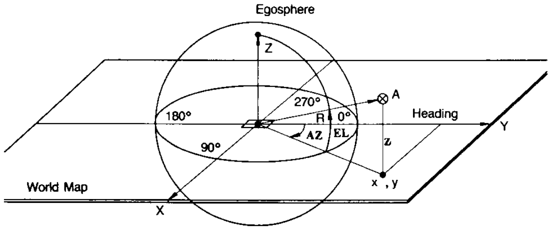


Fig. 11. Geometric relationship between world map and egosphere coordinates.

Any point in the world topological map can thus be projected onto the egosphere (and vice versa when R is known). Projections from the egosphere to the world map will leave blank those map pixels that cannot be observed from the center of the egosphere. Q.E.D.

There are 2 × 2 transformations of the form

$$(AZ, EL)^T = F(az, el)^T$$

and

$$(az, el)^T = F'(AZ, EL)^T$$

that can relate any map point (AZ, EL) on one egosphere to a map point (az,el) on another egosphere of the same origin. The radius R to any egosphere pixel is unchanged by the F and F' transformations between egosphere representations with the same origin.

As ego motion occurs (i.e., as the self object moves through the world), the egosphere moves relative to world coordinates, and points on the egocentric maps flow across their surfaces. Ego motion may involve translation, or rotation, or both; in a stationary world, or a world containing moving objects. If egomotion is known, range to all stationary points in the world can be computed from observed image flow; and once range to any stationary point in the world is known, its pixel motion on the egosphere can be predicted from knowledge of egomotion. For moving points, prediction of pixel motion on the egosphere requires additional knowledge of object motion.

F. Egosphere Coordinate Systems

The proposed world model contains four different types of egosphere coordinates:

1) Sensor Egosphere Coordinates: The sensor egosphere is defined by the sensor position and orientation, and moves as the sensor moves. For vision, the sensor egosphere is the coordinate system of the retina. The sensor egosphere has coordinates of azimuth (AZ) and elevation (EL) fixed in the sensor system (such as an eye or a TV camera), as shown in Fig. 12. For a narrow field of view, rows and columns (x, z) in a flat camera image array correspond quite closely to azimuth and elevation (AZ, EL) on the sensor egosphere. However, for a wide field of view, the egosphere and flat image array representations have widely different geometries. The flat image (x, z) representation becomes highly elongated for a wide field of view, going to infinity at plus and minus 90 degrees. The egosphere representation, in contrast, is well behaved over the entire sphere (except for singularities at the egosphere poles).

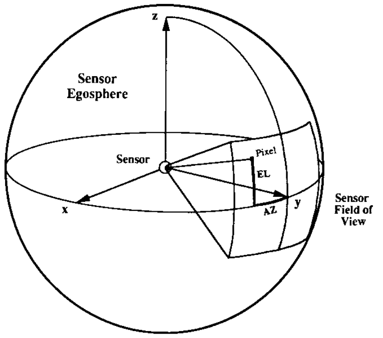


Fig. 12. Sensor egosphere coordinates. Azimuth (AZ) is measured clockwise from the sensor y-axis in the x–y plane. Elevation (EL) is measured up and down (plus and minus) from the x–y plane.

The sensor egosphere representation is useful for the analysis of wide angle vision such as occurs in the eyes of most biological creatures. For example, most insects and fish, many birds, and most prey animals such as rabbits have eyes with fields of view up to 180 degrees. Such eyes are often positioned on opposite sides of the head so as to provide almost 360 degree visual coverage. The sensor egosphere representation provides a tractable coordinate frame in which this type of vision can be analyzed.

2) Head Egosphere Coordinates: The head egosphere has (AZ, EL) coordinates measured in a reference frame fixed in the head (or sensor platform). The head egosphere representation is well suited for fusing sensory data from multiple sensors, each of which has its own coordinate system. Vision data from multiple eyes or cameras can be overlaid and registered in order to compute range from stereo. Directional and range data from acoustic and sonar sensors can be overlaid on vision data. Data derived from different sensors, or from multiple readings of the same sensor, can be overlaid on the head egosphere to build up a single image of multidimensional reality.

Pixel data in sensor egosphere coordinates can be transformed into the head egosphere by knowledge of the position and orientation of the sensor relative to the head. For example, the position of each eye in the head is fixed and the orientation of each eye relative to the head is known from stretch sensors in the ocular muscles. The position of tactile sensors relative to the head is known from proprioceptive sensors in the neck, torso, and limbs.

Hypothesis: Neuronal maps on the tectum (or superior colliculus), and on parts of the extrastriate visual cortex, are represented in a head egosphere coordinate system.

Receptive fields from the two retinas are well known to be overlaid in registration on the tectum, and superior colliculus. Experimental evidence indicates that registration and fusion of data from visual and auditory sensors takes place in the tectum of the barn owl [41] and the superior colliculus of the monkey [42] in head egosphere coordinates. Motor output for eye motion from the superior colliculus apparently is transformed

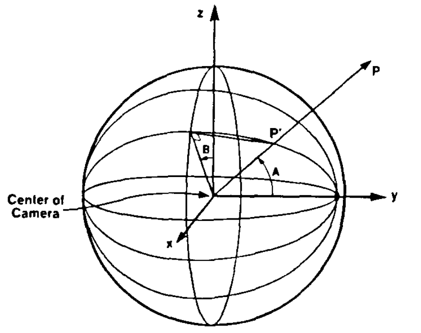


Fig. 13. The velocity egosphere. On the velocity egosphere, the y-axis is defined by the velocity factor. the x-axis points to the horizon on the right. A is the angle between the velocity vector and a pixel on the egosphere, and B is the angles between the z-axis and the plane defined by the velocity vector and the pixel vector.

back into retinal egosphere coordinates. There is also evidence that head egosphere coordinates are used in the visual areas of the parietal cortex [43], [54].

3) Velocity Egosphere: The velocity egosphere is defined by the velocity vector and the horizon. The velocity vector defines the pole (y-axis) of the velocity egosphere, and the x-axis points to the right horizon as shown in Fig. 13. The egosphere coordinates (A, B) are defined such that A is the angle between the pole and a pixel, and B is the angle between the yoz plane and the plane of the great circle flow line containing the pixel.

For egocenter translation without rotation through a stationary world, image flow occurs entirely along great circle arcs defined by B =constant. The positive pole of the velocity egosphere thus corresponds to the focus-of-expansion. The negative pole corresponds to the focus-of-contraction. The velocity egosphere is ideally suited for computing range from image flow, as discussed in Section XIV.

4) Inertial Egosphere: The inertial egosphere has coordinates of azimuth measured from a fixed point (such as North) on the horizon, and elevation measured from the horizon.

The inertial egosphere does not rotate as a result of sensor or body rotation. On the inertial egosphere, the world is perceived as stationary despite image motion due to rotation of the sensors and the head.

Fig. 14 illustrates the relationships between the four egosphere coordinate systems. Pixel data in eye (or camera) egosphere coordinates can be transformed into head (or sensor platform) egosphere coordinates by knowledge of the position and orientation of the sensor relative to the head. For example, the position of each eye in the head is fixed and the orientation of each eye relative to the head is known from stretch receptors in the ocular muscles (or pan and tilt encoders on a camera platform). Pixel data in head egosphere coordinates can be transformed into inertial egosphere coordinates by knowing the orientation of the head in inertial space. This information can be obtained from the vestibular (or inertial) system that measures the direction of gravity relative to the head and integrates rotary accelerations to obtain head position in inertial space. The inertial egosphere can be transformed into world coordinates by knowing the x, y, z position of the center of the egosphere. This is obtained from knowledge about where the self is located in the world. Pixels on any egosphere can be transformed into the velocity egosphere by knowledge of the direction of the current velocity vector on that egosphere. This can be obtained from a number of sources including the locomotion and vestibular systems.

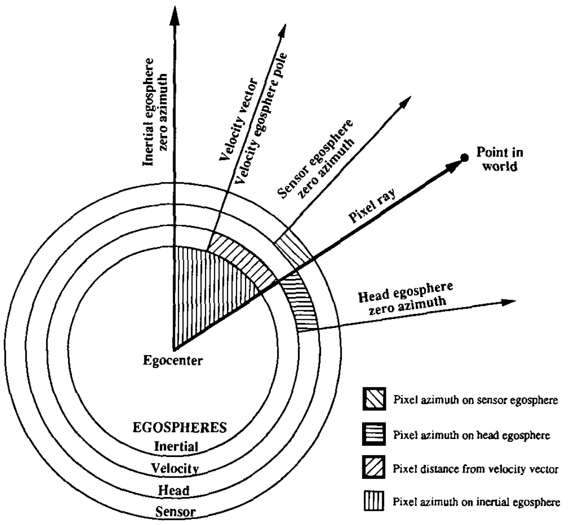

Inertial egosphere zero azimuth
Velocity vector
Velocity egosphere pole
Sensor egosphere zero azimuth
Point in world
Pixel ray
Head egosphere zero azimuth
Egocenter
EGOSPHERES
Inertial
Velocity
Head
Sensor
▧ Pixel azimuth on sensor egosphere
▤ Pixel azimuth on head egosphere
▨ Pixel distance from velocity vector
▥ Pixel azimuth on inertial egosphere

Fig. 14. A 2–D projection of four egosphere representations illustrating angular relationships between egospheres. Pixels are represented on each egosphere such that images remains in registration. Pixel attributes detected on one egosphere may thus be inherited on others. Pixel resolution is not typically uniform on a single egosphere, nor is it necessarily the same for different egospheres, or for different attributes on the same egosphere.

All of the previous egosphere transformations can be inverted, so that conversions can be made in either direction. Each transformation consists of a relatively simple vector function that can be computed for each pixel in parallel. Thus the overlay of sensory input with world model data can be accomplished in a few milliseconds by the type of computing architectures known to exist in the brain. In artificial systems, full image egosphere transformations can be accomplished within a television frame interval by state-of-the-art serial computing hardware. Image egosphere transformations can be accomplished in a millisecond or less by parallel hardware.

Hypothesis: The WM world maps, object maps, and egospheres are the brains data fusion mechanisms. They provide coordinate systems in which to integrate information from arrays of sensors (i.e., rods and cones in the eyes, tactile sensors in the skin, directional hearing, etc.) in space and time. They allow information from different sensory modalities (i.e., vision, hearing, touch, balance, and proprioception) to be combined into a single consistent model of the world.

Hypothesis: The WM functions that transform data between the world map and the various egosphere representations are the brain's geometry engine. They transform world model predictions into the proper coordinate systems for real-time comparison and correlation with sensory observations. This provides the basis for recognition and perception.

Transformations to and from the sensor egosphere, the inertial egosphere, the velocity egosphere, and the world map allow the intelligent system to sense the world from one perspective and interpret it in another. They allow the intelligent system to compute how entities in the world would look from another viewpoint. They provide the ability to overlay sensory input with world model predictions, and to compute the geometrical and dynamical functions necessary to navigate, focus attention, and direct action relative to entities and regions of the world.

G. Entities

Definition: An entity is an element from the set {point, line, surface, object, group}.

The world model contains information about entities stored in lists, or frames. The knowledge database contains a list of all the entities that the intelligent system knows about. A subset of this list is the set of current-entities known to be present in any given situation. A subset of the list of current-entities is the set of entities-of-attention.

There are two types of entities: generic and specific. A generic entity is an example of a class of entities. A generic entity frame contains the attributes of its class. A specific entity is a particular instance of an entity. A specific entity frame inherits the attributes of the class to which it belongs. An example of an entity frame might be:

ENTITY NAME	name of entity
kind	class or species of entity
type	generic or specific point, line, surface, object, or group
position	world map coordinates (uncertainty); egosphere coordinates (uncertainty)
dynamics	velocity (uncertainty);acceleration (uncertainty)
trajectory	sequence of positions
geometry	center of gravity (uncertainty); axis of symmetry (uncertainty);size (uncertainty);shape boundaries (uncertainty)
links	subentities; parent entity
properties	physical: mass; color; substance; behavioral: social (of animate objects)
capabilities	speed, range
value state-variables	attract-repulse; confidence-fear; love-hate

For example, upon observing a specific cow named Bertha, an entity frame in the brain of a visitor to a farm might have the following values:

ENTITY NAME	Bertha
kind	cow
type	specific object
position	x, y, z (in pasture map coordinates) AZ, EL, R (in egosphere image of observer)
dynamics	velocity, acceleration (in egosphere or pasture map coordinates)
trajectory	sequence of map positions while grazing
geometry	axis of symmetry (right/left) size (6 × 3 × 10 ft) shape (quadruped)
links	subentities - surfaces (torso, neck, head, legs, tail, etc.) parent entity - group (herd)
properties	physical:mass (1050 lbs); color (black and white); substance (flesh, bone, skin, hair); behavioral (standing, placid, timid, etc.)
capabilities	speed, range
value state-variables	attract-repulse = 3 (visitor finds cows moderately attractive) confidence-fear= -2 (visitor slightly afraid of cows) love-hate = 1 (no strong feelings)

H. Map–Entity Relationship

Map and entity representations are cross referenced and tightly coupled by real-time computing hardware. Each pixel on the map has in its frame a pointer to the list of entities covered by that pixel. For example, each pixel may cover a point entity indicating brightness, color, spatial and temporal gradients of brightness and color, image flow, and range for each point. Each pixel may also cover a linear entity indicating a brightness or depth edge or vertex; a surface entity indicating area, slope, and texture; an object entity indicating the name and attributes of the object covered; a group entity indicating the name and attributes of the group covered, etc.

Likewise, each entity in the attention list may have in its frame a set of geometrical parameters that enables the world model geometry engine to compute the set of egosphere or world map pixels covered by each entity, so that entity parameters associated with each pixel covered can be overlaid on the world and egosphere maps.

Cross referencing between pixel maps and entity frames allows the results of each level of processing to add map overlays to the egosphere and world map representations. The entity database can be updated from knowledge of image parameters at points on the egosphere, and the map database can be predicted from knowledge of entity parameters in the world model. At each level, local entity and map parameters can be computed in parallel by the type of neurological computing structures known to exist in the brain.

Many of the attributes in an entity frame are time dependent state-variables. Each time dependent state-variable may possess a short term memory queue wherein is stored a state trajectory, or trace, that describes its temporal history.

At each hierarchical level, temporal traces stretch backward about as far as the planning horizon at that level stretches into the future. At each hierarchical level, the historical trace of an entity state-variable may be captured by summarizing data values at several points in time throughout the historical interval. Time dependent entity state-variable histories may also be captured by running averages and moments, Fourier transform coefficients, Kalman filter parameters, or other analogous methods.

Each state-variable in an entity frame may have value state-variable parameters that indicate levels of believability, confidence, support, or plausibility, and measures of dimensional uncertainty. These are computed by value judgment functions that reside in the VJ modules. These are described in Section XV.

Value state-variable parameters may be overlaid on the map and egosphere regions where the entities to which they are assigned appear. This facilitates planning. For example, approach-avoidance behavior can be planned on an egosphere map overlay defined by the summation of attractor and repulsor value state-variables assigned to objects or regions that appear on the egosphere. Navigation planning can be done on a map overlay whereon risk and benefit values are assigned to regions on the egosphere or world map.

I. Entity Database Hierarchy

The entity database is hierarchically structured. Each entity consists of a set of subentities, and is part of a parent entity. For example, an object may consist of a set of surfaces, and be part of a group.

The definition of an object is quite arbitrary, however, at least from the point of view of the world model. For example, is a nose an object? If so, what is a face? Is a head an object? Or is it part of a group of objects comprising a body? If a body can be a group, what is a group of bodies?

Only in the context of a task, does the definition of an object become clear. For example, in a task frame, an object may be defined either as the agent, or as acted upon by the agent executing the task. Thus, in the context of a specific task, the nose (or face, or head) may become an object because it appears in a task frame as the agent or object of a task.

Perception in an intelligent system is task (or goal) driven, and the structure of the world model entity database is defined by, and may be reconfigured by, the nature of goals and tasks. It is therefore not necessarily the role of the world model to define the boundaries of entities, but rather to represent the boundaries defined by the task frame, and to map regions and entities circumscribed by those boundaries with sufficient resolution to accomplish the task. It is the role of the sensory processing system to identify regions and entities in the external real world that correspond to those represented in the world model, and to discover boundaries that circumscribe objects defined by tasks.

Theorem: The world model is hierarchically structured with map (iconic) and entity (symbolic) data structures at each level of the hierarchy.

At level 1, the world model can represent map overlays for point entities. In the case of vision, point entities may consist of brightness or color intensities, and spatial and temporal derivatives of those intensities. Point entity frames include brightness spatial and temporal gradients and range from stereo for each pixel. Point entity frames also include transform parameters to and from head egosphere coordinates. These representations are roughly analogous to Marr's "primal sketch" [44], and are compatible with experimentally observed data representations in the tectum, superior colliculus, and primary visual cortex ($V1$) [31].

At level 2, the world model can represent map overlays for linear entities consisting of clusters, or strings, of point entities. In the visual system, linear entities may consist of connected edges (brightness, color, or depth), vertices, image flow vectors, and trajectories of points in space/time. Attributes such as 3-D position, orientation, velocity, and rotation are represented in a frame for each linear entity. Entity frames include transform parameters to and from inertial egosphere coordinates. These representations are compatible with experimentally observed data representations in the secondary visual cortex (V2) [54].

At level 3, the world model can represent map overlays for surface entities computed from sets of linear entities clustered or swept into bounded surfaces or maps, such as terrain maps, B-spline surfaces, or general functions of two variables. Surface entities frames contain transform parameters to and from object coordinates. In the case of vision, entity attributes may describe surface color, texture, surface position and orientation, velocity, size, rate of growth in size, shape, and surface discontinuities or boundaries. Level 3 is thus roughly analogous to Marr's "2 1/2-D sketch", and is compatible with known representation of data in visual cortical areas V3 and V4.

At level 4, the world model can represent map overlays for object entities computed from sets of surfaces clustered or swept so as to define 3-D volumes, or objects. Object entity frames contain transform parameters to and from object coordinates. Object entity frames may also represent object type, position, translation, rotation, geometrical dimensions, surface properties, occluding objects, contours, axes of symmetry, volumes, etc. These are analogous to Marr's "3-D model" representation, and compatible with data representations in occipital-temporal and occipital-parietal visual areas.

At level 5, the world model can represent map overlays for group entities consisting of sets of objects clustered into groups or packs. This is hypothesized to correspond to data representations in visual association areas of parietal and temporal cortex. Group entity frames contain transform parameters to and from world coordinates. Group entity frames may also represent group species, center of mass, density, motion, map position, geometrical dimensions, shape, spatial axes of symmetry, volumes, etc.

At level 6, the world model can represent map overlays for sets of group entities clustered into groups of groups, or group^2 entities. At level 7, the world model can represent map overlays for sets of group^2 entities clustered into group^3 (or world) entities, and so on. At each higher level, world map resolution decreases and range increases by about an order of magnitude per level.

The highest level entity in the world model is the world itself, i.e., the environment as a whole. The environment entity frame contains attribute state-variables that describe the state of the environment, such as temperature, wind, precipitation, illumination, visibility, the state of hostilities or peace, the current level of danger or security, the disposition of the gods, etc.

J. Events

Definition: An event is a state, condition, or situation that exists at a point in time, or occurs over an interval in time.

Events may be represented in the world model by frames with attributes such as the point, or interval, in time and space when the event occurred, or is expected to occur. Event frames attributes may indicate start and end time, duration, type, relationship to other events, etc.

An example of an event frame is:

EVENT NAME	name of event
kind	class or species
type	generic or specific
modality	visual, auditory, tactile, etc.
time	when event detected
interval	period over which event took place
position	map location where event occurred
links	subevents; parent event
value	good-bad, benefit-cost, etc.

State-variables in the event frame may have confidence levels, degrees of support and plausibility, and measures of dimensional uncertainty similar to those in spatial entity frames. Confidence state-variables may indicate the degree of certainty that an event actually occurred, or was correctly recognized.

The event frame database is hierarchical. At each level of the sensory processing hierarchy, the recognition of a pattern, or string, of level(i) events makes up a single level($i+1$) event.

Hypothesis: The hierarchical levels of the event frame database can be placed in one-to-one correspondence with the hierarchical levels of task decomposition and sensory processing.

For example at: Level 1—an event may span a few milliseconds. A typical level(1) acoustic event might be the recognition of a tone, hiss, click, or a phase comparison indicating the direction of arrival of a sound. A typical visual event might be a change in pixel intensity, or a measurement of brightness gradient at a pixel.

Level 2—an event may span a few tenths of a second. A typical level(2) acoustic event might be the recognition of a phoneme or a chord. A visual event might be a measurement of image flow or a trajectory segment of a visual point or feature.

Level 3—an event may span a few seconds, and consist of the recognition of a word, a short phrase, or a visual gesture, or motion of a visual surface.

Level 4—an event may span a few tens of seconds, and consist of the recognition of a message, a melody, or a visual observation of object motion, or task activity.

Level 5—an event may span a few minutes and consist of listening to a conversation, a song, or visual observation of group activity in an extended social exchange.

Level 6—an event may span an hour and include many auditory, tactile, and visual observations.

Level 7—an event may span a day and include a summary of sensory observations over an entire day's activities.

XIV. Sensory Processing

Definition: Sensory processing is the mechanism of perception.

Theorem: Perception is the establishment and maintenance of correspondence between the internal world model and the external real world.

Corollary: The function of sensory processing is to extract information about entities, events, states, and relationships in the external world, so as keep the world model accurate and up to date.

A. Measurement of Surfaces

World model maps are updated by sensory measurement of points, edges, and surfaces. Such information is usually derived from vision or touch sensors, although some intelligent systems may derive it from sonar, radar, or laser sensors.

The most direct method of measuring points, edges, and surfaces is through touch. Many creatures, from insects to mammals, have antennae or whiskers that are used to measure the position of points and orientation of surfaces in the environment. Virtually all creatures have tactile sensors in the skin, particularly in the digits, lips, and tongue. Proprioceptive sensors indicate the position of the feeler or tactile sensor relative to the self when contact is made with an external surface. This, combined with knowledge of the kinematic position of the feeler endpoint, provides the information necessary to compute the position on the egosphere of each point contacted. A series of felt points defines edges and surfaces on the egosphere.

Another primitive measure of surface orientation and depth is available from image flow (i.e., motion of an image on the retina of the eye). Image flow may be caused either by motion of objects in the world, or by motion of the eye through the world. The image flow of stationary objects caused by translation of the eye is inversely proportional to the distance from the eye to the point being observed. Thus, if eye rotation is zero, and the translational velocity of the eye is known, the focus of expansion is fixed, and image flow lines are defined by great circle arcs on the velocity egosphere that emanate from the focus of expansion and pass through the pixel in question [45]. Under these conditions, range to any stationary point in the world can be computed directly from image flow by the simple formula

$$R = \frac{v \sin A}{dA/dt} \tag{1}$$

where R is the range to the point, v is translational velocity vector of the eye, A is the angle between the velocity vector and the pixel covering the point. dA/dt is the image flow rate at the pixel covering the point

When eye rotation is zero and v is known, the flow rate dA/dt can be computed locally for each pixel from temporal and spatial derivatives of image brightness along flow lines on the velocity egosphere. dA/dt can also be computed from temporal crosscorrelation of brightness from adjacent pixels along flow lines.

When the eye fixates on a point, dA/dt is equal to the rotation rate of the eye. Under this condition, the distance to the fixation point can be computed from (1), and the distance to other points may be computed from image flow relative to the fixation point.

If eye rotation is nonzero but known, the range to any stationary point in the world may be computed by a closed form formula of the form

$$R = F\left(x, y, T, W, \frac{\partial I}{\partial t}, \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) \tag{2}$$

where x and z are the image coordinates of a pixel, T is the translational velocity vector of the camera in camera coordinates, W is the rotational velocity vector of the camera in camera coordinates, and I is the pixel brightness intensity. This type of function can be implemented locally and in parallel by a neural net for each image pixel [46].

Knowledge of eye velocity, both translational and rotational, may be computed by the vestibular system, the locomotion system, and/or high levels of the vision system. Knowledge of, rotational eye motion may either be used in the computation of range by (2), or can be used to transform sensor egosphere images into velocity egosphere coordinates where (1) applies. This can be accomplished mechanically by the vestibulo-ocular reflex, or electronically (or neuronally) by scrolling the input image through an angle determined by a function of data variables from the vestibular system and the ocular muscle stretch receptors. Virtual transformation of image coordinates can be accomplished using coordinate transform parameters located in each map pixel frame.

Depth from image flow enables creatures of nature, from fish and insects to birds and mammals, to maneuver rapidly through natural environments filled with complex obstacles without collision. Moving objects can be segmented from stationary by their failure to match world model predictions for stationary objects. Near objects can be segmented from distant by their differential flow rates.

Distance to surfaces may also be computed from stereovision. The angular disparity between images in two eyes separated by a known distance can be used to compute range. Depth from stereo is more complex than depth from image flow in that it requires identification of corresponding points in images from different eyes. Hence it cannot be computed locally. However, stereo is simpler than image flow in that it does not require eye translation and is not confounded by eye rotation or by moving objects in the world. The computation of distance from a combination of both motion and stereo is more robust, and hence psychophysically more vivid to the observer, than from either motion or stereo alone.

Distance to surfaces may also be computed from sonar or radar by measuring the time delay between emitting radiation and receiving an echo. Difficulties arise from poor angular resolution and from a variety of sensitivity, scattering, and multipath problems. Creatures such as bats and marine mammals use multispectral signals such as chirps and clicks to minimize confusion from these effects. Phased arrays and synthetic apertures may also be used to improve the resolution of radar or sonar systems.

All of the previous methods for perceiving surfaces are primitive in the sense that they compute depth directly from sensory input without recognizing entities or understanding anything about the scene. Depth measurements from primitive processes can immediately generate maps that can be used directly by the lower levels of the behavior generation hierarchy to avoid obstacles and approach surfaces.

Surface attributes such as position and orientation may also be computed from shading, shadows, and texture gradients. These methods typically depend on higher levels of visual perception such as geometric reasoning, recognition of objects, detection of events and states, and the understanding of scenes.

B. Recognition and Detection

Definition: Recognition is the establishment of a one-to-one match, or correspondence, between a real world entity and a world model entity.

The process of recognition may proceed top-down, or bottom-up, or both simultaneously. For each entity in the world model, there exists a frame filled with information that can be used to predict attributes of corresponding entities observed in the world. The top-down process of recognition begins by hypothesizing a world model entity and comparing its predicted attributes with those of the observed entity. When the similarities and differences between predictions from the world model and observations from sensory processing are integrated over a space-time window that covers an entity, a matching, or crosscorrelation value is computed between the entity and the model. If the correlation value rises above a selected threshold, the entity is said to be recognized. If not, the hypothesized entity is rejected and another tried.

The bottom-up process of recognition consists of applying filters and masks to incoming sensory data, and computing image properties and attributes. These may then be stored in the world model, or compared with the properties and attributes of entities already in the world model. Both top-down and bottom-up processes proceed until a match is found, or the list of world model entities is exhausted. Many perceptual matching processes may operate in parallel at multiple hierarchical levels simultaneously.

If a SP module recognizes a specific entity, the WM at that level updates the attributes in the frame of that specific WM entity with information from the sensory system.

If the SP module fails to recognize a specific entity, but instead achieves a match between the sensory input and a generic world model entity, a new specific WM entity will be created with a frame that initially inherits the features of the generic entity. Slots in the specific entity frame can then be updated with information from the sensory input.

If the SP module fails to recognize either a specific or a generic entity, the WM may create an "unidentified" entity with an empty frame. This may then be filled with information gathered from the sensory input.

When an unidentified entity occurs in the world model, the behavior generation system may (depending on other priorities) select a new goal to <identify the unidentified entity>. This may initiate an exploration task that positions and focuses the sensor systems on the unidentified entity, and possibly even probes and manipulates it, until a world model frame is constructed that adequately describes the entity. The sophistication and complexity of the exploration task depends on task knowledge about exploring things. Such knowledge may be very advanced and include sophisticated tools and procedures, or very primitive. Entities may, of course, simply remain labeled as "unidentified," or unexplained.

Event detection is analogous to entity recognition. Observed states of the real world are compared with states predicted by the world model. Similarities and differences are integrated over an event space-time window, and a matching, or crosscorrelation value is computed between the observed event and the model event. When the crosscorrelation value rises above a given threshold, the event is detected.

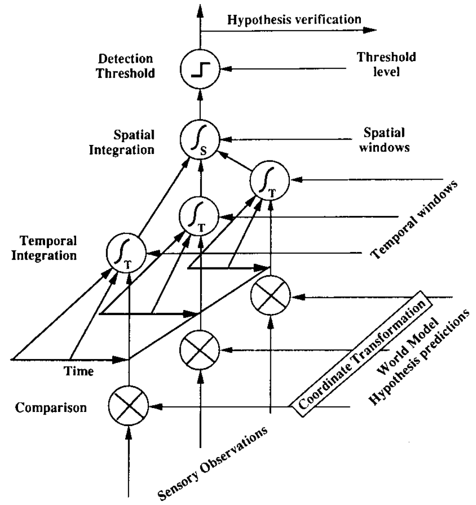


Fig. 15. Each sensory processing SP module consists of the following. 1) A set of comparators that compare sensory observations with world model predictions, 2) a set of temporal integrators that integrate similarities and differences, 3) a set of spatial integrators that fuse information from different sensory data streams, and 4) a set of threshold detectors that recognize entities and detect events.

C. The Context of Perception

If, as suggested in Fig. 5, there exists in the world model at every hierarchical level a short term memory in which is stored a temporal history consisting of a series of past values of time dependent entity and event attributes and states, it can be assumed that at any point in time, an intelligent system has a record in its short term memory of how it reached its current state. Figs. 5 and 6 also imply that, for every planner in each behavior generating BG module at each level, there exists a plan, and that each executor is currently executing the first step in its respective plan. Finally, it can be assumed that the knowledge in all these plans and temporal histories, and all the task, entity, and event frames referenced by them, is available in the world model.

Thus it can be assumed that an intelligent system almost always knows where it is on a world map, knows how it got there, where it is going, what it is doing, and has a current list of entities of attention, each of which has a frame of attributes (or state variables) that describe the recent past, and provide a basis for predicting future states. This includes a prediction of what objects will be visible, where and how object surfaces will appear, and which surface boundaries, vertices, and points will be observed in the image produced by the sensor system. It also means that the position and motion of the eyes, ears, and tactile sensors relative to surfaces and objects in the world are known, and this knowledge is available to be used by the sensory processing system for constructing maps and overlays, recognizing entities, and detecting events.

Were the aforementioned not the case, the intelligent system would exist in a situation analogous to a person who suddenly awakens at an unknown point in space and time. In such cases, it typically is necessary even for humans to perform a series of tasks designed to "regain their bearings", i.e., to bring their world model into correspondence with the state of the external world, and to initialize plans, entity frames, and system state variables.

It is, of course, possible for an intelligent creature to function in a totally unknown environment, but not well, and not for long. Not well, because every intelligent creature makes much good use of the historical information that forms the context of its current task. Without information about where it is, and what is going on, even the most intelligent creature is severely handicapped. Not for long, because the sensory processing system continuously updates the world model with new information about the current situation and its recent historical development, so that, within a few seconds, a functionally adequate map and a usable set of entity state variables can usually be acquired from the immediately surrounding environment.

D. Sensory Processing SP Modules

At each level of the proposed architecture, there are a number of computational nodes. Each of these contains an SP module, and each SP module consists of four sublevels, as shown in Fig. 15.

Sublevel 1—Comparison: Each comparison submodule matches an observed sensory variable with a world model prediction of that variable. This comparison typically involves an arithmetic operation, such as multiplication or subtraction, which yields a measure of similarity and difference between an observed variable and a predicted variable. Similarities indicate the degree to which the WM predictions are correct, and hence are a measure of the correspondence between the world model and reality. Differences indicate a lack of correspondence between world model predictions and sensory observations. Differences imply that either the sensor data or world model is incorrect. Difference images from the comparator go three places:

1) They are returned directly to the WM for real-time local pixel attribute updates. This produces a tight feedback loop whereby the world model predicted image becomes an array of Kalman filter state-estimators. Difference images are thus error signals by which each pixel of the predicted image can be trained to correspond to current sensory input.
2) They are also transmitted upward to the integration sublevels where they are integrated over time and space in order to recognize and detect global entity attributes. This integration constitutes a summation, or chunking, of sensory data into entities. At each level, lower order entities are "chunked" into higher order entities, i.e., points are chunked into lines, lines into surfaces, surfaces into objects, objects into groups, etc.
3) They are transmitted to the VJ module at the same level where statistical parameters are computed in order to assign confidence and believability factors to pixel entity attribute estimates.

Sublevel 2—Temporal integration: Temporal integration submodules integrate similarities and differences between predictions and observations over intervals of time. Temporal integration submodules operating just on sensory data can produce a summary, such as a total, or average, of sensory information over a given time window. Temporal integrator submodules operating on the similarity and difference values computed by comparison submodules may produce temporal crosscorrelation and covariance functions between the model and the observed data. These correlation and covariance functions are measures of how well the dynamic properties of the world model entity match those of the real world entity. The boundaries of the temporal integration window may be derived from world model prediction of event durations, or form behavior generation parameters such as sensor fixation periods.

Sublevel 3—Spatial integration: Spatial integrator submodules integrate similarities and differences between predictions and observations over regions of space. This produces spatial crosscorrelation or convolution functions between the model and the observed data. Spatial integration summarizes sensory information from multiple sources at a single point in time. It determines whether the geometric properties of a world model entity match those of a real world entity. For example, the product of an edge operator and an input image may be integrated over the area of the operator to obtain the correlation between the image and the edge operator at a point. The limits of the spatial integration window may be determined by world model predictions of entity size. In some cases, the order of temporal and spatial integration may be reversed, or interleaved.

Sublevel 4—Recognition/Detection threshold: When the spatiotemporal correlation function exceeds some threshold, object recognition (or event detection) occurs. For example, if the spatiotemporal summation over the area of an edge operator exceeds threshold, an edge is said to be detected at the center of the area.

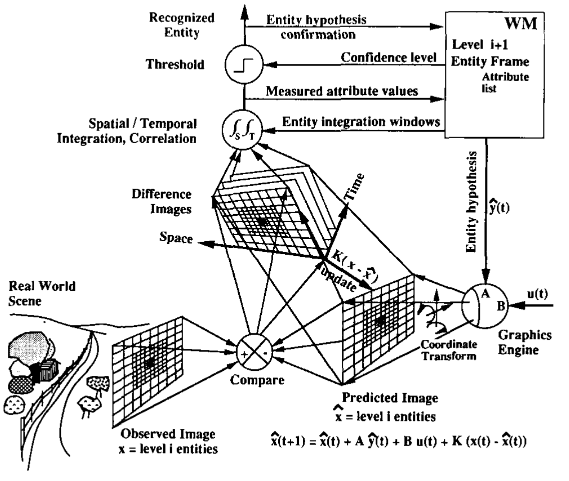


Fig. 16. Interaction between world model and sensory processing. Difference images are generator by comparing predicted images with observed images. Feedback of differences produces a Kalman best estimate for each data variable in the world model. Spatial and temporal integration produce crosscorrelation functions between the estimated attributes in the world model and the real-world attributes measured in the observed image. When the correlation exceeds threshold, entity recognition occurs.

Fig. 16 illustrates the nature of the SP-WM interactions between an intelligent vision system and the world model at one level. On the left of Fig. 16, the world of reality is viewed through the window of an egosphere such as exists in the primary visual cortex. On the right is a world model consisting of: 1) a symbolic entity frame in which entity attributes are stored, and 2) an iconic predicted image that is registered in real-time with the observed sensory image. In the center of Fig. 16, is a comparator where the expected image is subtracted from (or otherwise compared with) the observed image.

The level(i) predicted image is initialized by the equivalent of a graphics engine operating on symbolic data from frames of entities hypothesized at level($i+1$). The predicted image is updated by differences between itself and the observed sensory input. By this process, the predicted image becomes the world model's "best estimate prediction" of the incoming sensory image, and a high speed loop is closed between the WM and SP modules at level(i).

When recognition occurs in level (i), the world model level(i + 1) hypothesis is confirmed and both level(i) and level(i + 1) symbolic parameters that produced the match are updated in the symbolic database. This closes a slower, more global, loop between WM and SP modules through the symbolic entity frames of the world model. Many examples of this type of looping interaction can be found in the model matching and model-based recognition literature [47]. Similar closed loop filtering concepts have been used for years for signal detection, and for dynamic systems modeling in aircraft flight control systems. Recently they have been applied to high speed visually guided driving of an autonomous ground vehicle [48].

The behavioral performance of intelligent biological creatures suggests that mechanisms similar to those shown in Figs. 15 and 16 exist in the brain. In biological or neural network implementations, SP modules may contain thousands, even millions, of comparison submodules, temporal and spatial integrators, and threshold submodules. The neuroanatomy of the mammalian visual system suggests how maps with many different overlays, as well as lists of symbolic attributes, could be processed in parallel in real-time. In such structures it is possible for multiple world model hypotheses to be compared with sensory observations at multiple hierarchical levels, all simultaneously.

E. World Model Update

Attributes in the world model predicted image may be updated by a formula of the form

$$\hat{\boldsymbol{x}}(t+1) = \hat{\boldsymbol{x}}(t) + \boldsymbol{A}\hat{\boldsymbol{y}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{K}(t)[\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)] \tag{3}$$

where $\hat{x}(t)$ is the best estimate vector of world model i-order entity attributes at time t, A is a matrix that computes the expected rate of change of $\hat{x}(t)$ given the current best estimate of the $i+1$ order entity attribute vector $\hat{y}(t)$, B is a matrix that computes the expected rate of change of $\hat{x}(t)$ due to external input $u(t)$, and $K(t)$ is a confidence factor vector for updating $\hat{x}(t)$. The value of $K(t)$ may be computed by a formula of the form

$$\boldsymbol{K}(t) = K_s(j,t)[1 - K_m(j,t)] \tag{4}$$

where $K_s(j,t)$ is the confidence in the sensory observation of the jth real world attribute $x(j,t)$ at time t, $0 \leq K_s(j,t) \leq 1$ $K_m(j,t)$ is the confidence in the world model prediction of the jth attribute at time t $0 \leq K_m(j,t) \leq 1$.

The confidence factors (K_m and K_s) in formula (4) may depend on the statistics of the correspondence between the world model entity and the real world entity (e.g. the number of data samples, the mean and variance of $[\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)]$, etc.). A high degree of correlation between $x(t)$ and $[\hat{x}(t)]$ in both temporal and spatial domains indicates that entities or events have been correctly recognized, and states and attributes of entities and events in the world model correspond to those in the real world environment. World model data elements that match observed sensory data elements are reinforced by increasing the confidence, or believability factor, $K_m(j,t)$ for the entity or state at location j in the world model attribute lists. World model entities and states that fail to match sensory observations have their confidence factors $K_m(j,t)$ reduced. The confidence factor $K_s(j,t)$ may be derived from the signal-to-noise ratio of the jth sensory data stream.

The numerical value of the confidence factors may be computed by a variety of statistical methods such Baysian or Dempster–Shafer statistics.

F. The Mechanisms of Attention

Theorem: Sensory processing is an active process that is directed by goals and priorities generated in the behavior generating system.

In each node of the intelligent system hierarchy, the behavior generating BG modules request information needed for the current task from sensory processing SP modules. By means of such requests, the BG modules control the processing of sensory information and focus the attention of the WM and SP modules on the entities and regions of space that are important to success in achieving behavioral goals. Requests by BG modules for specific types of information cause SP modules to select particular sensory processing masks and filters to apply to the incoming sensory data. Requests from BG modules enable the WM to select which world model data to use for predictions, and which prediction algorithm to apply to the world model data. BG requests also define which correlation and differencing operators to use, and which spatial and temporal integration windows and detection thresholds to apply.

Behavior generating BG modules in the attention subsystem also actively point the eyes and ears, and direct the tactile sensors of antennae, fingers, tongue, lips, and teeth toward objects of attention. BG modules in the vision subsystem control the motion of the eyes, adjust the iris and focus, and actively point the fovea to probe the environment for the visual information needed to pursue behavioral goals [49], [50]. Similarly, BG modules in the auditory subsystem actively direct the ears and tune audio filters to mask background noises and discriminate in favor of the acoustic signals of importance to behavioral goals.

Because of the active nature of the attention subsystem, sensor resolution and sensitivity is not uniformly distributed, but highly focused. For example, receptive fields of optic nerve fibers from the eye are several thousand times more densely packed in the fovea than near the periphery of the visual field. Receptive fields of touch sensors are also several thousand times more densely packed in the finger tips and on the lips and tongue, than on other parts of the body such as the torso.

The active control of sensors with nonuniform resolution has profound impact on the communication bandwidth, computing power, and memory capacity required by the sensory processing system. For example, there are roughly 500 000 fibers in the the optic nerve from a single human eye. These fibers are distributed such that about 100 000 are concentrated in the ±1.0 degree foveal region with resolution of about 0.007 degrees. About 100 000 cover the surrounding ±3 degree region with resolution of about 0.02 degrees. 100 000 more cover the surrounding ±10 degree region with resolution of 0.07 degrees. 100 000 more cover the surrounding 30 degree region with a resolution of about 0.2 degrees. 100 000 more cover the remaining ±80 degree region with resolution of about 0.7 degree [51]. The total number of pixels is thus about 500 000 pixels, or somewhat less than that contained in two standard commercial TV images. Without nonuniform resolution, covering the entire visual field with the resolution of the fovea would require the number of pixels in about 6 000 standard TV images. Thus, for a vision sensory processing system with any given computing capacity, active control and nonuniform resolution in the retina can produce more than three orders of magnitude improvement in image processing capability.

SP modules in the attention subsystem process data from low-resolution wide-angle sensors to detect regions of interest, such as entities that move, or regions that have discontinuities (edges and lines), or have high curvature (corners and intersections). The attention BG modules then actively maneuver the eyes, fingers, and mouth so as to bring the high resolution portions of the sensory systems to bear precisely on these points of attention. The result gives the subjective effect of high resolution everywhere in the sensory field. For example, wherever the eye looks, it sees with high resolution, for the fovea is always centered on the item of current interest.

The act of perception involves both sequential and parallel operations. For example, the fovea of the eye is typically scanned sequentially over points of attention in the visual field [52]. Touch sensors in the fingers are actively scanned over surfaces of objects, and the ears may be pointed toward sources of sound. While this sequential scanning is going on, parallel recognition processes hypothesize and compare entities at all levels simultaneously.

G. The Sensory Processing Hierarchy

It has long been recognized that sensory processing occurs in a hierarchy of processing modules, and that perception proceeds by "chunking", i.e., by recognizing patterns, groups, strings, or clusters of points at one level as a single feature, or point in a higher level, more abstract space. It also has been observed that this chunking process proceeds by about an order of magnitude per level, both spatially and temporally [17], [18]. Thus, at each level in the proposed architecture, SP modules integrate, or chunk, information over space and time by about an order of magnitude.

Fig. 17 describes the nature of the interactions hypothesized to take place between the sensory processing and world modeling modules at the first four levels, as the recognition process proceeds. The functional properties of the SP modules are coupled to, and determined by, the predictions of the WM modules in their respective processing nodes. The WM predictions are, in turn, effected by states of the BG modules.

Hypothesis: There exist both iconic (maps) and symbolic (entity frames) at all levels of the SP/WM hierarchy of the mammalian vision system.

Fig. 18 illustrates the concept stated in this hypothesis. Visual input to the retina consists of photometric brightness and color intensities measured by rods and cones. Brightness intensities are denoted by $I(k, AZ, EL, t)$, where I is the brightness intensity measured at time t by the pixel at sensor egosphere azimuth AZ and elevation EL of eye (or camera) k. Retinal intensity signals I may vary over time intervals on the order of a millisecond or less.

Image preprocessing is performed on the retina by horizontal, bipolar, amacrine, and ganglion cells. Center-surround receptive fields ("on-center" and "off-center") detect both spatial and temporal derivatives at each point in the visual field. Outputs from the retina carried by ganglion cell axons become input to sensory processing level 1 as shown in Fig. 18. Level 1 inputs correspond to events of a few milliseconds duration.

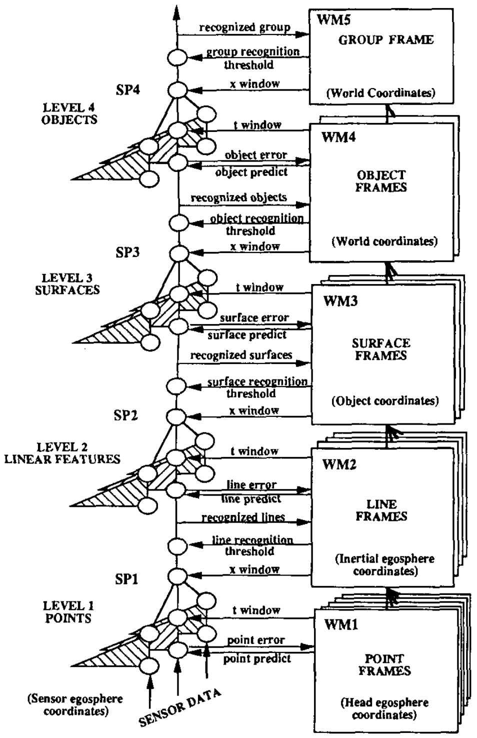


Fig. 17. The nature of the interactions that take place between the world model and sensory processing modules. At each level, predicted entities are compared with bo observed. Differences are returned as errors directly to the world model to update the model. Correlations are forwarded upward to be integrated over time and space windows provided by the world model. Correlations that exceed threshold are d recognized as entities.

It is hypothesized that in the mammalian brain, the level 1 vision processing module consists of the neurons in the lateral geniculate bodies, the superior colliculus, and the primary visual cortex ($V1$). Optic nerve inputs from the two eyes are overlaid such that the visual fields from left and right eyes are in registration. Data from stretch sensors in the ocular muscles provides information to the superior colliculus about eye convergence, and pan, tilt, and roll of the retina relative to the head. This allows image map points in retinal coordinates to be transformed into image map points in head coordinates (or vice versa) so that visual and acoustic position data can be registered and fused [41], [42]. In $V1$, registration of corresponding pixels from two separate eyes on single neurons also provides the basis for range from stereo to be computed for each pixel [31].

At level 1, observed point entities are compared with predicted point entities. Similarities and differences are integrated into linear entities. Strings of level 1 input events are integrated into level 1 output events spanning a few tens of milliseconds. Level 1 outputs become level 2 inputs.

The level 2 vision processing module is hypothesized to

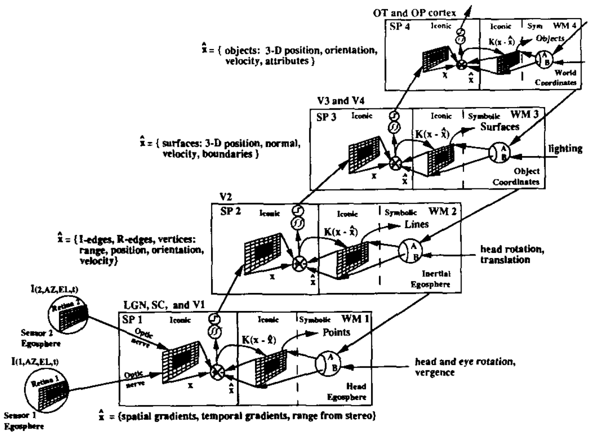


Fig. 18. Hypothesized correspondence between levels in the proposed model and neuranatomical structures in the mammalian vision system. At each level, the WM module contains both iconic and symbolic representations. At each level, the SP module compares the observed image with a predicted image. At each level, both iconic and symbolic world models are updated, and map overlays are computed. LGN is the lateral geniculate nuclei, OT is the occipital–temporal, OP is the occipital–parietal, and SC is the superior colliculus.

consist of neurons in the secondary visual cortex (V2). At level 2, observed linear entities are compared with predicted linear entities. Similarities and differences are integrated into surface entities. Some individual neurons indicate edges and lines at particular orientations. Other neurons indicate edge points, curves, trajectories, vertices, and boundaries.

Input to the world model from the vestibular system indicates the direction of gravity and the rotation of the head. This allows the level 2 world model to transform head egosphere representations into inertial egosphere coordinates where the world is perceived to be stationary despite rotation of the sensors.

Acceleration data from the vestibular system, combined with velocity data from the locomotion system, provide the basis for estimating both rotary and linear eye velocity, and hence image flow direction. This allows the level 2 world model to transform head egosphere representations into velocity egosphere coordinates where depth from image flow can be computed. Center-surround receptive fields along image flow lines can be subtracted from each other to derive spatial derivatives in the flow direction. At each point where the spatial derivative in the flow direction is nonzero, spatial and temporal derivatives can be combined with knowledge of eye velocity to compute the image flow rate dA/dt [45]. Range to each pixel can then be computed directly, and in parallel, from local image data using formula (1) or (2).

The previous egosphere transformations do not necessarily imply that neurons are physically arranged in inertial or velocity egosphere coordinates on the visual cortex. If that were true, it would require that the retinal image be scrolled over the cortex, and there is little evidence for this, at least in $V1$ and $V2$. Instead, it is conjectured that the neurons that make up both observed and predicted iconic images exist on the visual cortex in retinotopic, or sensor egosphere, coordinates. The velocity and inertial egosphere coordinates for each pixel are defined by parameters in the symbolic entity frame of each pixel. The inertial, velocity (and perhaps head) egospheres may thus be "virtual" egospheres. The position of any pixel on any egosphere can be computed by using the transformation parameters in the map pixel frame as an indirect address offset. This allows velocity and inertial egosphere computations to be performed on neural patterns that are physically represented in sensor egosphere coordinates.

The possibility of image scrolling cannot be ruled out, however, particularly at higher levels. It has been observed that both spatial and temporal retinotopic specificity decreases about two orders of magnitude from $V1$ to $V4$ [54]. This is consistent with scrolling.

Strings of level 2 input events are integrated into level 3 input events spanning a few hundreds of milliseconds.

The level 3 vision processing module is hypothesized to reside in areas $V3$ and $V4$ of the visual cortex. Observed surface entities are compared with predicted surface entities. Similarities and differences are integrated to recognize object entities. Cells that detect texture and motion of regions in specific directions provide indication of surface boundaries and depth discontinuities. Correlations and differences between world model predictions and sensory observations of surfaces give rise to meaningful image segmentation and recognition of surfaces. World model knowledge of lighting and texture allow computation of surface orientation, discontinuities, boundaries, and physical properties.

Strings of level 3 input events are integrated into level 4 input events spanning a few seconds. (This does not necessarily imply that it takes seconds to recognize surfaces, but that both patterns of motion that occupy a few seconds, and surfaces, are recognized at level 3. For example, the recognition of a gesture, or dance step, might occur at this level.)

World model knowledge of the position of the self relative to surfaces enables level 3 to compute offset variables for each pixel that transform it from inertial egosphere coordinates into object coordinates.

The level 4 vision processing module is hypothesized to reside in the posterior inferior temporal and ventral intraparietal regions of visual cortex. At level 4, observed objects are compared with predicted objects. Correlations and differences between world model predictions and sensory observations of objects allows shape, size, and orientation, as well as location, velocity, rotation, and size-changes of objects to be recognized and measured.

World model input from the locomotion and navigation systems allow level 4 to transform object coordinates into world coordinates. Strings of level 4 input events are grouped into level 5 input events spanning a few tens of seconds.

Level 5 vision is hypothesized to reside in the visual association areas of the parietal and temporal cortex. At level 5, observed groups of objects are compared with predicted groups. Correlations are integrated into group^2 entities. Strings of level 5 input events are detected as level 5 output events spanning a few minutes. For example, in the anterior inferior temporal region particular groupings of objects such as eyes, nose, and mouth are recognized as faces. Groups of fingers can be recognized as hands, etc. In the parietal association areas, map positions, orientations, rotations of groups of objects are detected. At level 5, the world model map has larger span and lower resolution than level 4.

At level 6, clusters of group^2 entities are recognized as group^3 entities, and strings of level 6 input events are grouped into level 6 output events spanning a few tens of minutes. The world model map at level 7 has larger span and lower resolution than at level 6.

At level 7, strings of level 7 input events are grouped into level 7 output events spanning a few hours.

It must be noted that the neuroanatomy of the mammalian vision system is much more convoluted than suggested by Fig. 18. Van Essen [53] has compiled a list of 84 identified or suspected pathways connecting 19 visual areas. Visual processing is accomplished in at least two separate subsystems that are not differentiated in Fig. 18. The subsystem that includes the temporal cortex emphasizes the recognition of entities and their attributes such as shape, color, orientation, and grouping of features. The subsystem that includes the parietal cortex emphasizes spatial and temporal relationships such as map positions, timing of events, velocity, and direction of motion [54]. It should also be noted that analogous figures could be drawn for other sensory modalities such as hearing and touch.

H. Gestalt Effects

When an observed entity is recognized at a particular hierarchical level, its entry into the world model provides predictive support to the level below. The recognition output also flows upward where it narrows the search at the level above. For example, a linear feature recognized and entered into the world model at level 2, can be used to generate expected points at level 1. It can also be used to prune the search tree at level 3 to entities that contain that particular type of linear feature. Similarly, surface features at level 3 can generate specific expected linear features at level 2, and limit the search at level 4 to objects that contain such surfaces, etc. The recognition of an entity at any level thus provides to both lower and higher levels information that is useful in selecting processing algorithms and setting spatial and temporal integration windows to integrate lower level features into higher level chunks.

If the correlation function at any level falls below threshold, the current world model entity or event at that level will be rejected, and others tried. When an entity or event is rejected, the rejection also propagates both upward and downward, broadening the search space at both higher and lower levels.

At each level, the SP and WM modules are coupled so as to form a feedback loop that has the properties of a relaxation process, or phase-lock loop. WM predictions are compared with SP observations, and the correlations and differences are fed back to modify subsequent WM predictions. WM predictions can thus be "servoed" into correspondence with the SP observations. Such looping interactions will either converge to a tight correspondence between predictions and observations, or will diverge to produce a definitive set of irreconcilable differences.

Perception is complete only when the correlation functions at all levels exceed threshold simultaneously. It is the nature of closed loop processes for lock-on to occur with a positive "snap". This is especially pronounced in systems with many coupled loops that lock on in quick succession. The result is a gestalt "aha" effect that is characteristic of many human perceptions.

I. Flywheeling, Hysteresis, and Illusion

Once recognition occurs, the looping process between SP and WM acts as a tracking filter. This enables world model predictions to track real world entities through noise, data dropouts, and occlusions.

In the system described previously, recognition will occur when the first hypothesized entity exceeds threshold. Once recognition occurs, the search process is suppressed, and the thresholds for all competing recognition hypotheses are effectively raised. This creates a hysteresis effect that tends to keep the WM predictions locked onto sensory input during the tracking mode. It may also produce undesirable side effects, such as a tendency to perceive only what is expected, and a tendency to ignore what does not fit preconceived models of the world.

In cases where sensory data is ambiguous, there is more than one model that can match a particular observed object. The first model that matches will be recognized, and other models will be suppressed. This explains the effects produced by ambiguous figures such as the Necker cube.

Once an entity has been recognized, the world model projects its predicted appearance so that it can be compared with the sensory input. If this predicted information is added to, or substituted for, sensory input, perception at higher levels will be based on a mix of sensory observations and world model predictions. By this mechanism, the world model may fill in sensory data that is missing, and provide information that may be left out of the sensory data. For example, it is well known that the audio system routinely "flywheels" through interruptions in speech data, and fills-in over noise bursts.

This merging of world model predictions with sensory observations may account for many familiar optical illusions such as subjective contours and the Ponzo illusion. In pathological cases, it may also account for visions and voices, and an inability to distinguish between reality and imagination. Merging of world model prediction with sensory observation is what Grossberg calls "adaptive resonance" [55].

XV. Value Judgments

Value judgments provide the criteria for making intelligent choices. Value judgments evaluate the costs, risks, and benefits of plans and actions, and the desirability, attractiveness, and uncertainty of objects and events. Value judgment modules produce evaluations that can be represented as value state-variables. These can be assigned to the attribute lists in entity frames of objects, persons, events, situations, and regions of space. They can also be assigned to the attribute lists of plans and actions in task frames. Value state-variables can label entities, tasks, and plans as good or bad, costly or inexpensive, as important or trivial, as attractive or repulsive, as reliable or uncertain. Value state-variables can also be used by the behavior generation modules both for planning and executing actions. They provide the criteria for decisions about which coarse of action to take [56].

Definition: Emotions are biological value state-variables that provide estimates of good and bad.

Emotion value state-variables can be assigned to the attribute lists of entities, events, tasks, and regions of space so as to label these as good or bad, as attractive or repulsive, etc. Emotion value state-variables provide criteria for making decisions about how to behave in a variety of situations. For example, objects or regions labeled with fear can be avoided, objects labeled with love can be pursued and protected, those labeled with hate can be attacked, etc. Emotional value judgments can also label tasks as costly or inexpensive, risky or safe.

Definition: Priorities are value state-variables that provide estimates of importance.

Priorities can be assigned to task frames so that BG planners and executors can decide what to do first, how much effort to spend, how much risk is prudent, and how much cost is acceptable, for each task.

Definition: Drives are value state-variables that provide estimates of need.

Drives can be assigned to the self frame, to indicate internal system needs and requirements. In biological systems, drives indicate levels of hunger, thirst, and sexual arousal. In mechanical systems, drives might indicate how much fuel is left, how much pressure is in a boiler, how many expendables have been consumed, or how much battery charge is remaining.

A. The Limbic System

In animal brains, value judgment functions are computed by the limbic system. Value state-variables produced by the limbic system include emotions, drives, and priorities. In animals and humans, electrical or chemical stimulation of specific limbic regions (i.e., value judgment modules) has been shown to produce pleasure and pain as well as more complex emotional feelings such as fear, anger, joy, contentment, and despair. Fear is computed in the posterior hypothalamus. Anger and rage are computed in the amygdala. The insula computes feelings of contentment, and the septal regions produce joy and elation. The perifornical nucleus of the hypothalamus computes punishing pain, the septum pleasure, and the pituitary computes the body's priority level of arousal in response to danger and stress [57].

The drives of hunger and thirst are computed in the limbic system's medial and lateral hypothalamus. The level of sexual arousal is computed by the anterior hypothalamus. The control of body rhythms, such as sleep-awake cycles, are computed by the pineal gland. The hippocampus produces signals that indicate what is important and should be remembered, or what is unimportant and can safely be forgotten. Signals from the hippocampus consolidate (i.e., make permanent) the storage of sensory experiences in long term memory. Destruction of the hippocampus prevents memory consolidation [58].

In lower animals, the limbic system is dominated by the sense of smell and taste. Odor and taste provides a very simple and straight forward evaluation of many objects. For example, depending on how something smells, one should either eat it, fight it, mate with it, or ignore it. In higher animals, the limbic system has evolved to become the seat of much more sophisticated value judgments, including human emotions and appetites. Yet even in humans, the limbic system retains its primitive function of evaluating odor and taste, and there remains a close connection between the sense of smell and emotional feelings.

Input and output fiber systems connect the limbic system to sources of highly processed sensory data as well as to high level goal selection centers. Connections with the frontal cortex suggests that the value judgment modules are intimately involved with long range planning and geometrical reasoning. Connections with the thalamus suggests that the limbic value judgment modules have access to high level perceptions about objects, events, relationships, and situations; for example, the recognition of success in goal achievement, the perception of praise or hostility, or the recognition of gestures of dominance or submission. Connections with the reticular formation suggests that the limbic VJ modules are also involved in computing confidence factors derived from the degree of correlation between predicted and observed sensory input. A high degree of correlation produces emotional feelings of confidence. Low correlation between predictions and observations generates feelings of fear and uncertainty.

The limbic system is an integral and substantial part of the brain. In humans, the limbic system consists of about 53 emotion, priority, and drive submodules linked together by 35 major nerve bundles [57].

B. Value State-Variables

It has long been recognized by psychologists that emotions play a central role in behavior. Fear leads to flight, hate to rage and attack. Joy produces smiles and dancing. Despair produces withdrawal and despondent demeanor. All creatures tend to repeat what makes them feel good, and avoid what they dislike. All attempt to prolong, intensify, or repeat those activities that give pleasure or make the self feel confident, joyful, or happy. All try to terminate, diminish, or avoid those activities that cause pain, or arouse fear, or revulsion.

It is common experience that emotions provide an evaluation of the state of the world as perceived by the sensory system. Emotions tell us what is good or bad, what is attractive or repulsive, what is beautiful or ugly, what is loved or hated, what provokes laughter or anger, what smells sweet or rotten, what feels pleasurable, and what hurts.

It is also widely known that emotions affect memory. Emotionally traumatic experiences are remembered in vivid detail for years, while emotionally nonstimulating everyday sights and sounds are forgotten within minutes after they are experienced.

Emotions are popularly believed to be something apart from intelligence—irrational, beyond reason or mathematical analysis. The theory presented here maintains the opposite. In this model, emotion is a critical component of biological intelligence, necessary for evaluating sensory input, selecting goals, directing behavior, and controlling learning.

It is widely believed that machines cannot experience emotion, or that it would be dangerous, or even morally wrong to attempt to endow machines with emotions. However, unless machines have the capacity to make value judgments (i.e., to evaluate costs, risks, and benefits, to decide which course of action, and what expected results, are good, and which are bad) machines can never be intelligent or autonomous. What is the basis for deciding to do one thing and not another, even to turn right rather than left, if there is no mechanism for making value judgments? Without value judgments to support decision making, nothing can be intelligent, be it biological or artificial.

Some examples of value state-variables are listed below, along with suggestions of how they might be computed. This list is by no means complete.

Good is a high level positive value state-variable. It may be assigned to the entity frame of any event, object, or person. It can be computed as a weighted sum, or spatiotemporal integration, of all other positive value state-variables assigned to the same entity frame.

Bad is a high level negative value state-variable. It can be computed as a weighted sum, or spatiotemporal integration, of all other negative value state-variables assigned to an entity frame.

Pleasure: Physical pleasure is a mid-level internal positive value state-variable that can be assigned to objects, events, or specific regions of the body. In the latter case, pleasure may be computed indirectly as a function of neuronal sensory inputs from specific regions of the body. Emotional pleasure is a high level internal positive value state-variable that can be computed as a function of highly processed information about situations in the world.

Pain: Physical pain is a low level internal negative value state-variable that can be assigned to specific regions of the body. It may be computed directly as a function of inputs from pain sensors in specific regions of the body. Emotional pain is a high level internal negative value state-variable that may be computed indirectly from highly processed information about situations in the world.

Success_observed is a positive value state-variable that represents the degree to which task goals are met, plus the amount of benefit derived therefrom.

Success_expected is a value state-variable that indicates the degree of expected success (or the estimated probability of success). It may be stored in a task frame, or computed during planning on the basis of world model predictions. When compared with success_observed it provides a base-line for measuring whether goals were met on, behind, or ahead of schedule; at, over, or under estimated costs; and with resulting benefits equal to, less than, or greater than those expected.

Hope is a positive value state-variable produced when the world model predicts a future success in achieving a good situation or event. When high hope is assigned to a task frame, the BG module may intensify behavior directed toward completing the task and achieving the anticipated good situation or event.

Frustration is a negative value state-variable that indicates an inability to achieve a goal. It may cause a BG module to abandon an ongoing task, and switch to an alternate behavior. The level of frustration may depend on the priority attached to the goal, and on the length of time spent in trying to achieve it.

Love is a positive value state-variable produced as a function of the perceived attractiveness and desirability of an object or person. When assigned to the frame of an object or person, it tends to produce behavior designed to approach, protect, or possess the loved object or person.

Hate is a negative value state-variable produced as a function of pain, anger, or humiliation. When assigned to the frame of an object or person, hate tends to produce behavior designed to attack, harm, or destroy the hated object or person.

Comfort is a positive value state-variable produced by the absence of (or relief from) stress, pain, or fear. Comfort can be assigned to the frame of an object, person, or region of space that is safe, sheltering, or protective. When under stress or in pain, an intelligent system may seek out places or persons with entity frames that contain a large comfort value.

Fear is a negative value state-variable produced when the sensory processing system recognizes, or the world model predicts, a bad or dangerous situation or event. Fear may be assigned to the attribute list of an entity, such as an object, person, situation, event, or region of space. Fear tends to produce behavior designed to avoid the feared situation, event, or region, or flee from the feared object or person.

Joy is a positive value state-variable produced by the recognition of an unexpectedly good situation or event. It is assigned to the self-object frame.

Despair is a negative value state-variable produced by world model predictions of unavoidable, or unending, bad situations or events. Despair may be caused by the inability of the behavior generation planners to discover an acceptable plan for avoiding bad situations or events.

Happiness is a positive value state-variable produced by sensory processing observations and world model predictions of good situations and events. Happiness can be computed as a function of a number of positive (rewarding) and negative (punishing) value state-variables.

Confidence is an estimate of probability of correctness. A confidence state-variable may be assigned to the frame of any entity in the world model. It may also be assigned to the self frame, to indicate the level of confidence that a creature has in its own capabilities to deal with a situation. A high value of confidence may cause the BG hierarchy to behave confidently or aggressively.

Uncertainty is a lack of confidence. Uncertainty assigned to the frame of an external object may cause attention to be directed toward that object in order to gather more information about it. Uncertainty assigned to the self-object frame may cause the behavior generating hierarchy to be timid or tentative.

It is possible to assign a real nonnegative numerical scalar value to each value state-variable. This defines the degree, or amount, of that value state-variable. For example, a positive real value assigned to "good" defines how good; i.e., if

$$e := \text{"good" and } 0 \leq e \leq 10 \qquad (5)$$

then, $e = 10$ is the "best" evaluation possible.

Some value state-variables can be grouped as conjugate pairs. For example, good-bad, pleasure-pain, success-fail, love-hate, etc. For conjugate pairs, a positive real value means the amount of the good value, and a negative real value means the amount of the bad value.

For example, if

$$e := \text{"good-bad" and } -10 \leq e \leq +10$$

then $e = 5$ is good $e = 6$ is better $e = 10$ is best $e = -4$ is bad $e = -7$ is worse $e = -10$ is worst $e = 0$ is neither good nor bad.

Similarly, in the case of pleasure-pain, the larger the positive value, the better it feels. The larger the negative value, the worse it hurts. For example, if

$$e := \text{"pleasure-pain"}$$

then $e = 5$ is pleasurable $e = 10$ is ecstasy $e = -5$ is painful $e = -10$ is agony $e = 0$ is neither pleasurable nor painful.

The positive and negative elements of the conjugate pair may be computed separately, and then combined.

C. VJ Modules

Value state-variables are computed by value judgment functions residing in VJ modules. Inputs to VJ modules describe entities, events, situations, and states. VJ value judgment functions compute measures of cost, risk, and benefit. VJ outputs are value state-variables.

Theorem: The VJ value judgment mechanism can be defined as a mathematical or logical function of the form

$$\boldsymbol{E} = \boldsymbol{V}(\boldsymbol{S}) \qquad (6)$$

where $\boldsymbol{E}$ is an output vector of value state-variables, $\boldsymbol{V}$ is a value judgment function that computes $\boldsymbol{E}$ given $\boldsymbol{S}$, $\boldsymbol{S}$ is an input state vector defining conditions in the world model, including the self. The components of $\boldsymbol{S}$ are entity attributes describing states of tasks, objects, events, or regions of space. These may be derived either from processed sensory information, or from the world model.

The value judgment function $\boldsymbol{V}$ in the VJ module computes a numerical scalar value (i.e., an evaluation) for each component of $\boldsymbol{E}$ as a function of the input state vector $\boldsymbol{S}$, $\boldsymbol{E}$ is a time dependent vector. The components of $\boldsymbol{E}$ may be assigned to attributes in the world model frame of various entities, events, or states.

If time dependency is included, the function $\boldsymbol{E}(t + dt) = \boldsymbol{V}(\boldsymbol{S}(t))$ may be computed by a set of equations of the form

$$e(j, t + dt) = (k\, d/dt + 1) \sum_i s(i, t) w(i, j) \qquad (7)$$

where $e(j, t)$ is the value of the jth value state-variable in the vector $\boldsymbol{E}$ at time t $s(i, t)$ is the value of the ith input variable at time t $w(i, j)$ is a coefficient, or weight, that defines the contribution of $s(i)$ to $e(j)$.

Each individual may have a different set of "values", i.e., a different weight matrix in its value judgment function $\boldsymbol{V}$.

The factor $(kd/dt + 1)$ indicates that a value judgment is typically dependent on the temporal derivative of its input variables as well as on their steady-state values. If $k > 1$, then the rate of change of the input factors becomes more important than their absolute values. For $k > 0$, need reduction and escape from pain are rewarding. The more rapid the escape, the more intense, but short-lived, the reward.

Formula (8) suggests how a VJ function might compute the value state-variable "happiness":

$$\begin{aligned} \text{happiness} = (k\, d/dt + 1)(&\text{success-expectation} \\ &+ \text{hope-frustration} \\ &+ \text{love-hate} \\ &+ \text{comfort-fear} \\ &+ \text{joy-despair}) \end{aligned} \qquad (8)$$

where success, hope, love, comfort, joy are all positive value state-variables that contribute to happiness, and expectation, frustration, hate, fear, and despair are all negative value state-variables that tend to reduce or diminish happiness. In this example, the plus and minus signs result from $+1$ weights assigned to the positive-value state-variables, and -1 weights assigned to the negative-value state-variables. Of course, different brains may assign different values to these weights.

Expectation is listed in formula (8) as a negative state-variable because the positive contribution of success is diminished if success_observed does not meet or exceed success_expected. This suggests that happiness could be increased if expectations were lower. However, when $k > 0$, the hope reduction that accompanies expectation downgrading may be just as punishing as the disappointments that result from unrealistic expectations, at least in the short term. Therefore, lowering expectations is a good strategy for increasing happiness only if expectations are lowered very slowly, or are already low to begin with.

Fig. 19 shows an example of how a VJ module might compute pleasure-pain. Skin and muscle are known to contain arrays of pain sensors that detect tissue damage. Specific receptors for pleasure are not known to exist, but pleasure state-variables can easily be computed from intermediate state-variables that are computed directly from skin sensors.

The VJ module in Fig. 19 computes "pleasure-pain" as a function of the intermediate state-variables of "softness", "warmth", and "gentle stroking of the skin". These intermediate state-variables are computed by low level SP modules.

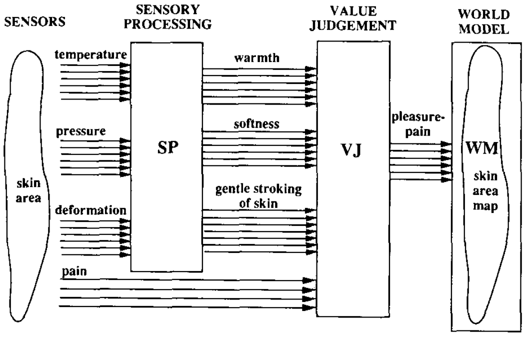


Fig. 19. How a VJ value judgment module might evaluate tactile and thermal sensory input. In this example, pleasure–pain is computed by a VJ module as a function of "warmth," "softness," and "gentle stroking" state-variables recognized by an SP module, plus inputs directly from pain sensors in the skin. Pleasure–pain value state-variables are assigned to pixel frames of the world model map of the skin area.

"warmth" is computed from temperature sensors in the skin. "softness" is computed as a function of "pressure" and "deformation" (i.e., stretch) sensors. "gentle stroking of the skin" is computed by a spatiotemporal analysis of skin pressure and deformation sensor arrays that is analogous to image flow processing of visual information from the eyes. Pain sensors go directly from the skin area to the VJ module.

In the processing of data from sensors in the skin, all of the computations preserve the topological mapping of the skin area. Warmth is associated with the area in which the temperature sensors are elevated. Softness is associated with the area where pressure and deformation are in the correct ratio. Gentle stroking is associated with the area in which the proper spatiotemporal patterns of pressure and deformation are observed. Pain is associated with the area where pain sensors are located. Finally, pleasure-pain is associated with the area from which the pleasure-pain factors originate. A pleasure-pain state-variable can thus be assigned to the knowledge frames of the skin pixels that lie within that area.

D. *Value State-Variable Map Overlays*

When objects or regions of space are projected on a world map or egosphere, the value state-variables in the frames of those objects or regions can be represented as overlays on the projected regions. When this is done, value state-variables such as comfort, fear, love, hate, danger, and safe will appear overlaid on specific objects or regions of space. BG modules can then perform path planning algorithms that steer away from objects or regions overlaid with fear, or danger, and steer toward or remain close to those overlaid with attractiveness, or comfort. Behavior generation may generate attack commands for target objects or persons overlaid with hate. Protect, or care-for, commands may be generated for target objects overlaid with love.

Projection of uncertainty, believability, and importance value state-variables on the egosphere enables BG modules to perform the computations necessary for manipulating sensors and focusing attention.

Confidence, uncertainty, and hope state-variables may also be used to modify the effect of other value judgments. For example, if a task goal frame has a high hope variable but low confidence variable, behavior may be directed toward the hoped-for goal, but cautiously. On the other hand, if both hope and confidence are high, pursuit of the goal may be much more aggressive.

The real-time computation of value state-variables for varying task and world model conditions provides the basis for complex situation dependent behavior [56].

XVI. Neural Computation

Theorem: All of the processes described previously for the BG, WM, SP, and VJ modules, whether implicit or explicit, can be implemented in neural net or connectionist architectures, and hence could be implemented in a biological neuronal substrate.

Modeling of the neurophysiology and anatomy of the brain by a variety of mathematical and computational mechanisms has been discussed in a number of publications [16], [27], [34], [35] ,[55], [59]–[64]. Many of the submodules in the BG, WM, SP, and VJ modules can be implemented by functions of the form $\boldsymbol{P}=\boldsymbol{H}(\boldsymbol{S})$. This type of computation can be accomplished directly by a typical layer of neurons that might make up a section of cortex or a subcortical nucleus.

To a first approximation,_any single neuron, such as illustrated in Fig. 20, can compute a linear single valued function of the form

$$p(k) = h(\boldsymbol{S}) = \sum_{i=1}^{N} s(i)w(i,k) \tag{9}$$

where $p(k)$ is the output of the kth neuron; $\boldsymbol{S} = (s(1), s(2), \dots s(i), \dots, s(N))$ is an ordered set of input variables carried by input fibers defining an input vector; $\boldsymbol{W} = (w(1,k), w(2,k), \dots w(i,k), \dots w(N,k)$ is an ordered set of synaptic weights connecting the N input fibers to the kth neuron; and $h(\boldsymbol{S})$ is the internal product between the input vector and the synaptic weight vector.

A set of neurons of the type illustrated in Fig. 20 can therefore compute the vector function

$$\boldsymbol{P} = \boldsymbol{H}(\boldsymbol{S}) \tag{10}$$

where $\boldsymbol{P} = (p(1), p(2), \dots p(k), \dots p(L))$ is an ordered set of output variables carried by output fibers defining an output vector.

Axon and dendrite interconnections between layers, and within layers, can produce structures of the form illustrated in Fig. 4. State driven switching functions produce structures such as illustrated in Figs. 2 and 3. It has been shown how such structures can produce behavior that is sensory-interactive, goal-directed, and value driven.

The physical mechanisms of computation in a neuronal computing module are produced by the effect of chemical activation on synaptic sites. These are analog parameters with time constants governed by diffusion and enzyme activity rates. Computational time constants can vary from milliseconds to minutes, or even hours or days, depending on the chemicals

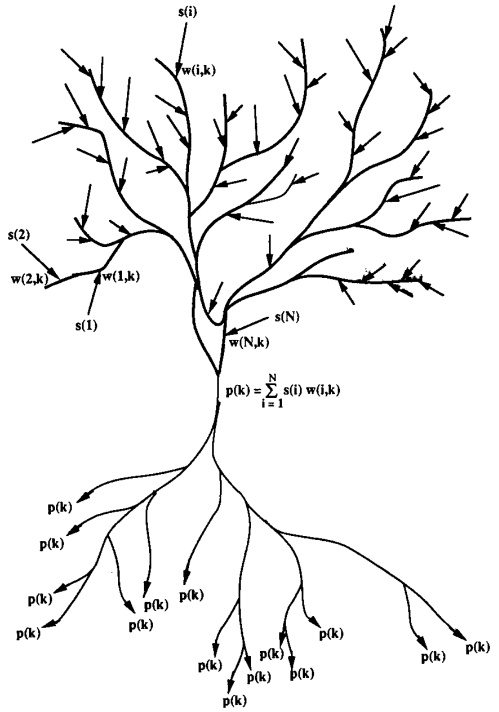


Fig. 20. A neuron computes the scalar value $p(k)$ as the inner product of the input vector $s(1), s(2), \ldots, s(i), \ldots, s(N)$ and the weight vector $w(1,k), w(2,k), \ldots w(i,k), \ldots, w(N,k)$.

carrying the messages, the enzymes controlling the decay time constants, the diffusion rates, and the physical locations of neurological sites of synaptic activity.

The time dependent functional relationship between input fiber firing vector $\boldsymbol{S}(t)$ and the output cell firing vector $\boldsymbol{P}(t)$ can be captured by making the neural net computing module time dependent

$$\boldsymbol{P}(t + dt) = \boldsymbol{H}(\boldsymbol{S}(t)). \tag{11}$$

The physical arrangement of input fibers in Fig. 20 can also produce many types of nonlinear interactions between input variables. It can, in fact, be shown that a computational module consisting of neurons of the type illustrated in Fig. 20 can compute any single valued arithmetic, vector, or logical function, IF/THEN rule, or memory retrieval operation that can be represented in the form $\boldsymbol{P}(t + dt) = \boldsymbol{H}(\boldsymbol{S}(t))$. By interconnecting $\boldsymbol{P}(t + dt) = \boldsymbol{H}(\boldsymbol{S}(t))$ computational modules in various ways, a number of additional important mathematical operations can be computed, including finite state automata, spatial and temporal differentiation and integration, tapped delay lines, spatial and temporal auto- and crosscorrelation, coordinate transformation, image scrolling and warping, pattern recognition, content addressable memory, and sampled-data, state-space feedback control. [59]–[63].

In a two layer neural net such as a Perceptron, or a brain model such as CMAC [27], [34], [35], the nonlinear function

$$\boldsymbol{P}(t + dt) = \boldsymbol{H}(\boldsymbol{S}(t))$$

is computed by a pair of functions

$$\boldsymbol{A}(\tau) = \boldsymbol{F}(\boldsymbol{S}(t)) \tag{12}$$

$$\boldsymbol{P}(t + dt) = \boldsymbol{G}(\boldsymbol{A}(\tau)) \tag{13}$$

where $\boldsymbol{S}(t)$ represents a vector of firing rates $s(i,t)$ on a set of input fibers at time t, $\boldsymbol{A}(\tau)$ represents a vector of firing rates $a(j,\tau)$ of a set of association cells at time $\tau = t + dt/2$, $\boldsymbol{P}(t + dt)$ represents a vector of firing rates $p(k, t + dt)$ on a set of output fibers at time $t + dt$, $\boldsymbol{F}$ is the function that maps S into $\boldsymbol{A}$, and $\boldsymbol{G}$ is the function that maps $\boldsymbol{A}$ into $\boldsymbol{P}$.

The function $\boldsymbol{F}$ is generally considered to be fixed, serving the function of an address decoder (or recoder) that transforms the input vector $\boldsymbol{S}$ into an association cell vector $\boldsymbol{A}$. The firing rate of each association cell $a(j,t)$ thus depends on the input vector $\boldsymbol{S}$ and the details of the interconnecting matrix of interneurones between the input fibers and association cells that define the function $\boldsymbol{F}$. Recoding from $\boldsymbol{S}$ to $\boldsymbol{A}$ can enlarge the number of patterns that can be recognized by increasing the dimensionality of the pattern space, and can permit the storage of nonlinear functions and the use of nonlinear decision surfaces by circumscribing the neighborhood of generalization. [34], [35].

The function $\boldsymbol{G}$ depends on the values of a set of synaptic weights $w(j,k)$ that connect the association cells to the output cells. The value computed by each output neuron $p(k,t)$ at time t is

$$p(k, t + dt) = \sum_{j} a(j) w(j,k) \tag{14}$$

where $w(j,k)$=synaptic weight from $a(j)$ to $p(k)$.

The weights $w(j,k)$ may be modified during the learning process so as to modify the function $\boldsymbol{G}$, and hence the function $\boldsymbol{H}$.

Additional layers between input and output can produce indirect addressing and list processing functions, including tree search and relaxation processes [16], [61]. Thus, virtually all of the computational functions required of an intelligent system can be produced by neuronal circuitry of the type known to exist in the brains of intelligent creatures.

XVII. Learning

It is not within the scope of this paper to review of the field of learning. However, no theory of intelligence can be complete without addressing this phenomenon. Learning is one of several processes by which world knowledge and task knowledge become embedded in the computing modules of an intelligent system. In biological systems, knowledge is also provided by genetic and growth mechanisms. In artificial systems, knowledge is most often provided through the processes of hardware engineering and software programming.

In the notation of (13), learning is the process of modifying the $\boldsymbol{G}$ function. This in turn, modifies the $\boldsymbol{P} = \boldsymbol{H}(\boldsymbol{S})$ functions that reside in BG, WM, SP, and VJ modules. Thus through learning, the behavior generation system can acquire new behavioral skills, the world model can be updated, the sensory processing system can refine its ability to interpret sensory input, and new parameters can be instilled in the value judgment system.

The change in strength of synaptic weights $w(j,k)$ wrought by the learning process may be described by a formula of the form

$$dw(j,k,t) = g(t)a(j,t)p(k,t) \tag{15}$$

where $dw(j,k,t)$ is the change in the synaptic weight $w(j,k,t)$ between t and $t+dt$; $g(t)$ is the learning gain at time t; $a(j,t)$ is the firing rate of association cell j at time t; and $p(k,t)$ is the firing rate of output neuron k at time t.

If $g(t)$ is positive, the effect will be to reward or strengthen active synaptic weights. If $g(t)$ is negative, the effect will be to punish, or weaken active synaptic weights.

After each learning experience, the new strength of synaptic weights is given by

$$w(j,k,t+dt) = w(j,k,t) + dw(j,k,t). \tag{16}$$

A. Mechanisms of Learning

Observations from psychology and neural net research suggests that there are at least three major types of learning: repetition, reinforcement, and specific error correction learning.

1) Repetition: Repetition learning occurs due to repetition alone, without any feedback from the results of action. For this type of learning, the gain function g is a small positive constant. This implies that learning takes place solely on the basis of coincidence between presynaptic and postsynaptic activity. Coincident activity strengthens synaptic connections and increases the probability that the same output activity will be repeated the next time the same input is experienced.

Repetition learning was first hypothesized by Hebb, and is sometimes called Hebbian learning. Hebb hypothesized that repetition learning would cause assemblies of cells to form associations between coincident events, thereby producing conditioning. Hebbian learning has been simulated in neural nets, with some positive results. However, much more powerful learning effects can be obtained with reinforcement learning.

2) Reinforcement: Reinforcement learning incorporates feedback from the results of action. In reinforcement learning, the learning gain factor $g(t)$ varies with time such that it conveys information as to whether the evaluation computed by the VJ module was good (rewarding), or bad (punishing). $g(t)$ is thus computed by a VJ function of the form

$$g(t+dt) = \boldsymbol{V}(\boldsymbol{S}(t)) \tag{17}$$

where $\boldsymbol{S}(t)$ is a time dependent state vector defining the object, event, or region of space being evaluated.

For task learning

$$g(t+dt) = \boldsymbol{V}\{\boldsymbol{R}(t) - \boldsymbol{R}_d(t)\} \tag{18}$$

where $\boldsymbol{R}(t)$ is the actual task results at time t, $\boldsymbol{R}_d(t)$ is the desired task results at time t, $\boldsymbol{R}(t) - \boldsymbol{R}_d(t)$ is the difference between the actual results and the desired results.

Task learning may modify weights in BG modules that define parameters in subtasks, or the weights that define decision functions in BG state-tables, or the value of state-variables in the task frame, such as task priority, expected cost, risk, or benefit. Task learning may thus modify both the probability that a particular task will be selected under certain conditions, and the way that the task is decomposed and executed when it is selected.

Attribute learning modifies weights that define state-variables in the attribute list of entity or event frames in the world model. Attribute learning was described earlier by (3) and (4).

For attribute learning

$$g(t+dt) = K_s(i,t)[1 - K_m(j,t)]\boldsymbol{V}(\text{attribute}_j) \tag{19}$$

where $K_s(i,t)$ is the degree of confidence in the sensory observation of the ith real world attribute at time t (See formula (4)); $K_m(j,t)$ is the degree of confidence in the prediction of the jth world model attribute at time t; and $V(\text{attribute}_j)$ is the importance of the jth world model attribute.

In general, rewarding reinforcement causes neurons with active synaptic inputs to increase the value or probability of their output the next time the same situation arises, or through generalization to increase the value or probability of their output the next time almost-the-same situation arises. Every time the rewarding situation occurs, the same synapses are strengthened, and the output (or its probability of occurring) is increased further.

For neurons in the goal selection portion of the BG modules, the rewarding reinforcement causes rewarding goals to be selected more often. Following learning, the probabilities are increased of EX submodules selecting next-states that were rewarded during learning. Similarly, the probabilities are increased of PL and JA submodules selecting plans that were successful, and hence rewarding, in the past.

For neurons in the WM modules, rewarding results following an action causes reward expectations to be stored in the frame of the task being executed. This leads to reward values being increased on nodes in planning graphs leading up to the rewarding results. Cost/benefit values placed in the frames of objects, events, and tasks associated with the rewarding results are also increased. As a result, the more rewarding the result of behavior, the more the behavior tends to be repeated.

Reward reinforcement learning in the BG system is a form of positive feedback. The more rewarding the task, the greater the probability that it will be selected again. The more it is selected, the more reward is produced and the more the tendency to select it is increased. This can drive the goal selection system into saturation, producing effects like addiction, unless some other process such as fatigue, boredom, or satiety produce a commensurate amount of negative $g(t)$ that is distributed over the population of weights being modified.

Punishing reinforcement, or error correcting, learning occurs when $g(t)$ is negative, i.e., punishing. In biological brains, error correction weakens synaptic weights that are active immediately prior to punishing evaluations from the emotional system. This causes the neurons activated by those synapses to decrease their output the next time the same situation arises. Every time the situation occurs and the punishing evaluation is given, the same synapses are weakened and the output (or its probability of occurring) is reduced.

For neurons in the goal selection portion of the BG modules, error correction tends to cause punishing tasks to be avoided. It decreases the probability of EX submodules selecting a punishing next state. It decreases the probability of PL and JA submodules selecting a punishing plan.

For neurons in the WM modules, punishment observed to follow an action causes punishment state variables to be inserted into the attribute list of the tasks, objects, events, or regions of space associated with the punishing feedback. Thus, punishment can be expected the next time the same action is performed on the same object, or the same event is encountered, or the same region of space is entered. Punishment expectations (i.e., fear) can be placed in the nodes of planning graphs leading to punishing task results. Thus, the more punishing the task, the more the task tends to be avoided.

Error correction learning is a form of negative feedback. With each training experience, the amount of error is reduced, and hence the amount of punishment. Error correction is therefore self limiting and tends to converge toward a stable result. It produces no tendencies toward addiction.

It does, however, reduce the net value of the synaptic weight pool. Without some other process such as excitement, or satisfaction, to generate a commensurate amount of reward reinforcement, there could result a reduction in stimulus to action, or lethargy.

3) Specific Error Correction Learning: In specific error correction, sometimes called teacher learning, not only is the overall behavioral result $g(t)$ known, but the correct or desired response $p_d(k,t)$ of each output neuron is provided by a teacher. Thus, the precise error $(p(k)-p_d(k))$ for each neuron is known. This correction can then be applied specifically to the weights of each neuron in an amount proportional to the direction and magnitude of the error of that neuron. This can be described by

$$dw(j,k,t) = g(t)a(j,t)(p(k,t) - p_d(k,t)) \qquad (20)$$

where $p_d(k,t)$ is the desired firing rate of neuron k at t and $-1 \leq g(t) < 0$.

Teacher learning tends to converge rapidly to stable precise results because it has knowledge of the desired firing rate for each neuron. Teacher learning is always error correcting. The teacher provides the correct response, and anything different is an error. Therefore, $g(t)$ must always be negative to correct the error. A positive $g(t)$ would only tend to increase the error.

If the value of $g(t)$ is set to -1, the result is one-shot learning. One-shot learning is learning that takes only one training cycle to achieve perfect storage and recall. One-shot teacher learning is often used for world model map and entity attribute updates. The SP module produces an observed value for each pixel, and this becomes the desired value to be stored in a world model map. A SP module may also produce observed values for entity attributes. These become desired values to be stored in the world model entity frame.

Teacher learning may also be used for task skill learning in cases where a high level BG module can act as a teacher to a lower level BG module, i.e., by providing desired output responses to specific command and feedback inputs.

It should be noted that, even though teacher learning may be one-shot, task skill learning by teacher may require many training cycles, because there may be very many ways that a task can be perturbed from its ideal performance trajectory. The proper response to all of these must be learned before the task skill is fully mastered. Also, the teacher may not have full access to all the sensory input going to the BG module that is being taught. Thus, the task teacher may not always be fully informed, and therefore may not always generate the correct desired responses.

Since teacher learning is punishing, it must be accompanied by some reward reinforcement to prevent eventually driving synaptic weights to zero. There is some evidence, that both reward reinforcement, and teacher learning, take place simultaneously in the cerebellum. Reward signals are thought to be carried by diffuse noradrenergic fibers that affect many thousands of neurons in the same way, while error correction signals are believed to be carried by climbing fibers each of which specifically targets a single neuron or a very small groups of neurons [27].

It should be noted, however, that much of the evidence for neuronal learning is ambiguous, and the exact mechanisms of learning in the brain are still uncertain. The very existence of learning in particular regions of the brain (including the cerebellum) is still controversial [65]. In fact, most of the interesting questions remain unanswered about how and where learning occurs in the neural substrate, and how learning produces all the effects and capabilities observed in the brain.

There are also many related questions as to the relationships between learning, instinct, imprinting, and the evolution of behavior in individuals and species.

XVIII. Conclusion

The theory of intelligence presented here is only an outline. It is far from complete. Most of the theorems have not been proven. Much of what has been presented is hypothesis and argument from analogy. The references cited in the bibliography are by no means a comprehensive review of the subject, or even a set of representative pointers into the literature. They simply support specific points. A complete list of references relevant to a theory of intelligence would fill a volume of many hundreds of pages. Many important issues remain uncertain and many aspects of intelligent behavior are unexplained.

Yet, despite its incomplete character and hypothetical nature, the proffered theory explains a lot. It is both rich and self consistent, but more important, it brings together concepts from a wide variety of disciplines into a single conceptual framework. There is no question of the need for a unifying theory. The amount of research currently underway is huge, and progress is rapid in many individual areas. Unfortunately, positive results in isolated fields of research have not coalesced into commensurate progress toward a general understanding of the nature of intelligence itself, or even toward improved abilities to build intelligent machine systems. Intelligent systems research is seriously impeded because of the lack of a widely accepted theoretical framework. Even a common definition of terms would represent a major step forward.

The model presented here only suggests how the neural substrate could generate the phenomena of intelligence, and how computer systems might be designed so as to produce intelligent behavior in machines. No claim is made that the proposed architecture fully explains how intelligence actually is generated in the brain. Natural intelligence is almost certainly generated in a great variety of ways, by a large number of mechanisms. Only a few of the possibilities have been suggested here.

The theory is expressed almost entirely in terms of explicit representations of the functionality of BG, WM, SP, and VJ modules. This almost certainly is not the way the brains of lower forms, such as insects, generate intelligent behavior. In simple brains, the functionality of planning, representing space, modeling and perceiving entities and events is almost surely represented implicitly, embedded in the specific connectivity of neuronal circuitry, and controlled by instinct.

In more sophisticated brains, however, functionality most likely is represented explicitly. For example, spatial information is quite probably represented in world and egosphere map overlays, and map pixels may indeed have frames. One of the principal characteristics of the brain is that the neural substrate is arranged in layers that have the topological properties of maps. Output from one layer of neurons selects, or addresses, sets of neurons in the next. This is a form a indirect addressing that can easily give rise to list structures, list processing systems, and object-oriented data structures. Symbolic information about entities, events, and tasks may very well be represented in neuronal list structures with the properties of frames. In some instances, planning probably is accomplished by searching game graphs, or by invoking rules of the form IF (S)/THEN (P).

Implicit representations have an advantage of simplicity, but at the expense of flexibility. Implicit representations have difficulty in producing adaptive behavior, because learning and generalization take place only over local neighborhoods in state-space. On the other hand, explicit representations are complex, but with the complexity comes flexibility and generality. Explicitly represented information is easily modified, and generalization can take place over entire classes of entities. Class properties can be inherited by subclasses, entity attributes can be modified by one-shot learning, and small changes in task or world knowledge can produce radically altered behavior. With explicit representations of knowledge and functionality, behavior can become adaptive, even creative.

This paper attempts to outline an architectural framework that can describe both natural and artificial implementations of intelligent systems. Hopefully, this framework will stimulate researchers to test its hypotheses, and correct its assumptions and logic where and when they are shown to be wrong. The near term goal should be to develop a theoretical model with sufficient mathematical rigor to support an engineering science of intelligent machine systems. The long term goal should be a full understanding of the nature of intelligence and behavior in both artificial and natural systems.

Acknowledgment

The author wishes to thank Alex Meystel, Robert Rosenfeld, John Simpson, Martin Herman, Ron Lumia, and Rick Quintero for their numerous helpful comments, and Cheryl Hutchins for her help in preparation of this manuscript. Funding to support this research has been derived from DARPA, NIST, NASA, Army, Navy, Air Force, and Bureau of Mines sources.

References

[1] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, D. R., "Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty," *Computer Survey*, vol. 23, pp. 213–253, June 1980.

[2] J. E. Laird, A. Newell, and P. Rosenbloom, "SOAR: An architecture for general intelligence," *Artificial Intell.*, vol. 33, pp. 1–64, 1987.

[3] Honeywell, Inc., "Intelligent Task Automation Interim Tech. Rep. II-4", Dec. 1987.

[4] J. Lowerie *et al.*, "Autonomous land vehicle," Annu. Rep., ETL-0413, Martin Marietta Denver Aerospace, July 1986.

[5] D. Smith and M. Broadwell, "Plan coordination in support of expert systems integration," in *Knowledge-Based Planning Workshop Proc.*, Austin, TX, Dec. 1987.

[6] J. R. Greenwood, G. Stachnick, H. S. Kaye, "A procedural reasoning system for army maneuver planning," in *Knowledge-Based Planning Workshop Proc.*, Austin, TX, Dec. 1987.

[7] A. J. Barbera, J. S. Albus, M. L. Fitzgerald, and L. S. Haynes, "RCS: The NBS real-time control system," in *Proc. Robots 8 Conf. Exposition*, Detroit, MI, June 1984.

[8] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics Automat.*, vol. RA-2, Mar. 1986.

[9] G. N. Saridis, "Foundations of the theory of intelligent controls," in *Proc. IEEE Workshop on Intelligent Contr.*, 1985.

[10] A. Meystel, "Intelligent control in robotics," *J. Robotic Syst.*, 1988.

[11] J. A. Simpson, R. J. Hocken, and J. S. Albus, "The Automated manufacturing research facility of the National Bureau of Standards," *J. Manufact. Syst.*, vol. 1, no. 1, 1983.

[12] J. S. Albus, C. McLean, A. J. Barbera, and M. L. Fitzgerald, "An architecture for real-time sensory-interactive control of robots in a manufacturing environment," presented at the 4th IFAC/IFIP Symp. on Inform. Contr. Problems in Manufacturing Technology, Gaithersburg, MD, Oct. 1982.

[13] J. S. Albus, "System description and design architecture for multiple autonomous undersea vehicles," Nat. Inst. Standards and Tech., Tech. Rep. 1251, Gaithersburg, MD, Sept. 1988.

[14] J. S. Albus, H. G. McCain, and R. Lumia, "NASA/NBS standard reference model for telerobot control system architecture (NASREM)" Nat. Inst. Standards and Tech., Tech. Rep. 1235, Gaithersburg, MD, 1989.

[15] B. Hayes-Roth, "A blackboard architecture for control," *Artificial Intell.*, pp. 252–321, 1985.

[16] J. S. Albus, *Brains, Behavior, and Robotics.* Peterbourough, NH: BYTE/McGraw-Hill, 1981.

[17] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psych. Rev.*, vol. 63, pp. 71–97, 1956.

[18] A. Meystel, "Theoretical foundations of planning and navigation for autonomous robots," *Int. J. Intelligent Syst.*, vol. 2, pp. 73–128, 1987

[19] M. Minsky, "A framework for representing knowledge," in *The Psychology of Computer Vision*, P. Winston, Ed. New York: McGraw-Hill, 1975, pp. 211–277.

[20] E. D. Sacerdoti, *A Structure for Plans and Behavior.* New York: Elsevier, 1977.

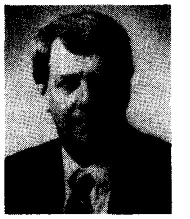
[21] R. C. Schank and R. P. Abelson, *Scripts Plans Goals and Understanding* Hillsdale, NJ: Lawrence Erlbaum, 1977.

[22] D. M. Lyons and M. A. Arbib, "Formal model of distributed computation sensory based robot control," *IEEE J. Robotics and Automat. Rev.*, 1988.

[23] D. W. Payton, "Internalized plans: A representation for action resources," *Robotics and Autonomous Syst.*, vol. 6, pp. 89–103, 1990.

[24] A. Sathi and M. Fox, "Constraint-directed negotiation of resource reallocations," CMU-RI-TR-89-12, Carnegie Mellon Robotics Institute Tech. Rep., Mar., 1989

[25] V. B. Brooks, *The Neural Basis of Motor Control.* Oxford, UK: Oxford Univ. Press, 1986.
[26] J. Piaget, *The Origins of Intelligence in Children.* New York: Int. Universities Press, 1952.
[27] J. S. Albus, "A theory of cerebellar function," *Math. Biosci.*, vol. 10, pp. 25–61, 1971.
[28] P. D. MacLean, *A Triune Concept of the Brain and Behavior.* Toronto, ON: Univ. Toronto Press, 1973.
[29] A. Schopenhauer, "The World As Will and Idea", 1883, in *The Philosophy of Schopenhauer*, Irwin Edman, Ed. Ithaca, NY: New York: Random House, 1928.
[30] J. J. Gibson, *The Ecological Approach to Visual Perception.* Ithaca, NY: Cornell Univ. Press, 1966.
[31] D. H. Hubel and T. N. Wiesel, "Ferrier lecture: Functional architecture of macaque monkey visual cortex," *Proc. Roy. Soc. Lond. B.* vol. 198, 1977, pp. 1–59.
[32] H. Samet, "The quadtree and related hierarchical data structures," *Computer Surveys*, pp. 16–2, 1984.
[33] P. Kinerva, *Sparse Distributed Memory.* Cambridge, MA: MIT Press, 1988.
[34] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Trans. ASME*, Sept. 1975.
[35] ——, "Data storage in the cerebellar model articulation controller (CMAC)," *Trans. ASME*, Sept. 1975.
[36] M. Bradey, "Computational approaches to image understanding," *ACM Comput. Surveys*, vol. 14, Mar. 1982.
[37] T. Binford, "Inferring surfaces from images," *Artificial Intell.*, vol. 17, pp. 205–244, 1981.
[38] D. Marr and H. K. Nishihara. "Representation and recognition of the spatial organization of three-dimensional shapes," in *Proc. Roy. Soc. Lond. B*, vol. 200, pp. 269–294, 1978.
[39] R. F. Riesenfeld, "Applications of B-spline approximation to geometric problems of computer aided design," Ph.D. dissertation, Syracuse Univ. 1973. available at Univ. Utah, UTEC -CSc-73-126.
[40] J. J. Koenderink, "The structure of images," *Biolog. Cybern.*, vol. 50, 1984.
[41] J. C. Pearson, J. Gelfand, W. Sullivan, R. Peterson, and C. Spence, "A neural network approach to sensory fusion," in *Proc. SPIE Sensor Fusion Conf.*, Orlando, FL, 1988.
[42] D. L. Sparks and M. Jay, "The role of the primate superior colliculus in sensorimotor integration," in *Vision, Brain, and Cooperative Computation*, Arbib and Hanson, Eds. Cambridge, MA: MIT Press, 1987.
[43] R. A. Andersen and D. Zipser, "The role of the posterior parietal cortex in coordinate transformations for visual-motor integration," *Can. J. Physiol. Pharmacol.*, vol. 66, pp. 488–501, 1988.
[44] D. Marr, *Vision.* San Francisco,CA: Freeman, 1982.
[45] J. S. Albus and T. H. Hong, "Motion, Depth, and Image Flow", in *Proc. IEEE Robotics and Automation*, Cincinnati, OH, 1990 (in process).
[46] D. Raviv and J. S. Albus, "Closed-form massively-parallel range-from-image flow algorithm," NISTIR 4450, National Inst. of Standards & Technology, Gaithersburg, MD, 1990.
[47] E. Kent and J. S. Albus, "Servoed world models as interfaces between robot control systems and sensory data," *Robotica*, vol. 2, pp. 17–25, 1984.
[48] E. D. Dickmanns and T. H. Christians, "Relative 3D-state estimation for autonomous visual guidance of road vehicles," *Intelligent Autonomous Syst.*, vol. 2, Amsterdam, The Netherlands, Dec. 11–14, 1989.
[49] R. Bajcsy, "Passive perception vs. active perception" in *Proc. IEEE Workshop on Computer Vision*, Ann Arbor, MI, 1986.
[50] K. Chaconas and M. Nashman, "Visual perception processing in a hierarchical control system: Level 1," Nat. Inst. Standards Technol. Tech. Note 1260, June 1989.
[51] Y. L. Grand, *Form and Space Vision*, Table 21, Ind. Univ. Press, Bloomington, IN, 1967.
[52] A. L. Yarbus, *Eye Movements and Vision.* New York: Plenum, 1967
[53] D. C. Van Essen, "Functional organization of primate visual cortex," *Cerebral Cortex*, vol. 3, A. Peters and E. G. Jones, Eds. New York: Plenum, 1985, pp. 259–329.
[54] J. H. R. Maunsell and W. T. Newsome, "Visual processing in monkey extrastriate cortex," *Ann. Rev. Neurosci.*, vol. 10, pp. 363–401, 1987.
[55] S. Grossberg, *Studies of Mind and Brain.* Amsterdam: Reidel, 1982.
[56] G. E. Pugh, *The Biological Origin of Human Values.* New York: Basic Books, 1977.
[57] A. C. Guyton, *Organ Physiology, Structure and Function of the Nervous System*, second ed. Philadelphia, PA: Saunders, 1976.
[58] W. B. Scoville and B. Milner, "Loss of recent memory after bilateral hippocampal lesions," *J. Neurophysiol. Neurosurgery Psychiatry*, vol. 20, no. 11, pp. 11–29, 1957.
[59] J. S. Albus, "Mechanisms of planning and problem solving in the brain," *Math. Biosci.*, vol. 45, pp. 247–293, 1979.
[60] S. Grossberg, Ed., *Neural Networks and Natural Intelligence.* Cambridge, MA: Bradford Books—MIT Press, 1988.
[61] J. S. Albus, "The cerebellum: A substrate for list-processing in the brain," in *Cybernetics, Artificial Intelligence and Ecology*, H. W. Robinson and D. E. Knight, Eds. Washington, DC: Spartan Books, 1972.
[62] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Acad. Sci.*, vol. 79, 1982, pp. 2554–2558.
[63] B. Widrow and R. Winter, "Neural nets for adpative filtering and adaptive pattern recognition," *Comput.*, vol. 21, no. 3, 1988.
[64] M. Minsky and S. Papert, *An Introduction to Computational Geometry.* Cambridge, MA: MIT Press, 1969.
[65] M. Ito, *The Cerebellum and Neuronal Control.* New York: Raven, 1984, ch. 10.



James S. Albus received the B. S. degree is physics in 1957 from Wheaton College, Wheaton, IL, the M.S. degree in electrical engineering in 1958 from Ohio State University, Columbus, and the Ph. D. degree in electrical engineering from the University of Maryland, College Park.

He is presently Chief of the Robot Systems Division, Laboartory for Manufacturing Engineering, National Institute of Standards and Technology, where he is responsible for robotics and automated manufacturing systems interface standards research. He designed the control system architecture for the Automated Manufacturing Research Facility. Previously, he worked from 1956 to 1973 for NASA Goddard Space Flight Center, where he designed electro-optical systems for more than 15 NASA spacecraft. For a short time, he served as program manager of the NASA Artificial Intelligence Program.

Dr. Albus has received several awards for his work in control theory, including the National Institute of Standards and Technology Applied Research Award, the Department of Commerce Gold and Silver Medals, The Industrial Research IR-100 award, and the Joseph F. Engelberger Award (which was presented at the International Robot Symposium in October 1984 by the King of Sweden. He has written two books, *Brains, Behavior, and Robotics* (Byte/McGraw Hill, 1981) and *Peoples' Capitalism: The Econimics of the Robot Revolutions* (New World Books, 1976). He is also the author of numerous scientific papers, journal articles, and official government studies, and has had articles published in *Scientific American*, *Omni*, *Byte*, and *Futurist.*