Sensing and Perception: Connectionist Approaches to "Subcognitive" Computing

J. Skrzypek
University of California, Los Angeles
Los Angeles, CA 90024

ABSTRACT

The machine perception laboratory represents a new paradigm for research in Artificial Intelligence at the Computer Science Department of UCLA. It is based on synergistic intermixing of methods and knowledge from the fields of Artificial Intelligence and Neuroscience.

-o- The Neuroscience is a source of fundamental concepts about function and mechanism of natural vision and perception; it motivates our view of inseparability between algorithms and neural substrate.

-o- The AI explores computational theories of vision and perceptual reasoning by inventing algorithms and implementing them as "connectionist" architectures.

The underlying intent of this interdisciplinary approach is to transform scientific knowledge into an engineering form of a general purpose machine perception by viewing "neural" connections as a paradigm for parallel computations.

The future of intelligent robots depends on succesfull implementation of a robust perceptual system. Although many clever forms of robotic vision have been engineered, a general-purpose machine perception remains a distant goal. Computing architectures best suited for global perceptual function pose one type of a problem. Another problem stems from the limitations of sequential computing paradigm where the number of functions which naturally map onto Von Neumann architecture is restricted. In natural system, visual functions are supported by a variety of parallel structures. This motivates our belief that future advances in a general purpose perception must assume inseparability of *function* from *strucuire*.

Our prototypical computational architecture consists of hierarchically structured layers of processing units that perform dedicated functions. Both discrete and real-value passing architectures are considered. Physical representation of transduced stimuli is implemented as a well structured connectivity between "neurons" and the computations are performed by types and weights of different connections. More precisely the computation is a result of some process, realized as "neuronal" functions, that is applied to a spatio-temporal "image" of signals. The process and the constraints are embedded into our connectionist architecture. The translation to more abstract levels is done through aggregation of features by an interpreter, which in early vision may be implemented by fixed connections. The ultimate goal of this project is to conceptualize a computing structure which could eventually be implemented in hardware.

Acknowledgement

This project was supported in part by NSF grant #ECS8307553, ARCO fellowship #1, Laboratory computing environment was made possible by a generous gift from Hewlett Packard and IBM. Thanks go to MPL members, David Gungner, Edmond Mesrobian, Paul Lin, Inge Heisey, Mike Suber and Eugene Paik for their suggestions and criticisms.

I. COMPUTERS AND BRAINS -- MOTIVATION

This paper is divided into four sections. First we outline the intellectual needs for integrating the knowledge about percepion in man and machine. The second section presents our notion of large grain architecture as a computational environment for studying global functions of machine perception. In the third part we describe the small grain architectures represented by "neural networks" that provide a computational substrate for perceptual functions. We conclude with architectural models of two early-vision operations implemented as neural networks that embody the *principle of inseparability between structure and function.*

a. Intellectual motivation

Intellectual motivations that unify studies of human and machine perception, - including vision, touch, proprioception, range and other sensory modalities, - derive from assumption that information processing is fundamental for intelligent behavior. Perception, spatial reasoning and learning are the attributes that will differentiate the next generation robots from present day automated manufacturing. The ultimate test for Artificial Intelligence is the invention of an autonomous mobile robots, whose "intelligent" behavior emerges from linking perception to motor output. Modern computer science plays a pivotal role in understanding information processing systems. On the other hand, mechanisms and functions of information processing underlying human intelligence are in the domain of Neurosciences. The rapid growth of these disciplines in recent years is advancing our understanding of perception. It is hoped that interdisciplinary combination of Artificial Intelligence and Cognitive Science will provide more rigorous, scientific fundations for this research.

What can be expected from a general theory of perception developed by such crossdisciplinary approach? In the short term it should help us understand how the elements of perception have evolved in natural systems and what are their limits. In the long run, a theory of perception should help us to formulate questions that extend beyond presently limited engineering knowledge of this function. For example, can we improve upon biological perception when implementing these functions in mobile robots? Is human perception limited by characteristics inherent only to biological systems? Are these limits imposed by algorithmic principles or by the underlying substrate? What is the grain of computing architecture most suitable for cognition and perception?

b. Perception and AI

Our working goal for Machine Perception and in particular for Computer Vision is a development of computing systems that can accomplish tasks previously only achieved with human intelligence (1). Discovery of heuristics used to constrain the problem according to physical laws should eventually lead to models of greater generality (2). In the past these efforts were strongly limited by the computational architectures available to the designer. The sequential computing paradigm limits solutions for computer vision that can operate in real time by restricting a selection of functions that naturally map onto Von Neuman architecture. In natural systems, visual functions are supported by a gamut of physical structures that are inherently masively parallel (3). Hence, we believe that further progress in realization of general purpose computer vision that operates in real ume must be based on assumption that function and the underlying computational substrate are inseparable. The chances of success can be maximized by combining traditional, forward-engineering approach to synthesis of computer vision system with analytic viewpoint as characterized by Neurosciences where the intent is to reverse engineer the solution. This is difficult because the current knowledge about anatomy and physiology of neuronal networks underlying manipulation of mental imagery does not allow easy introspection on such processes at the level of subcognitive computation (4, 5). Nevertheless, models of mental computation underlying perceptuon and cognition must be build and verified. Approximation of such tests at the present time, is possible only through computational models in the realm of AI (6). Our approach to studies of cognitive and perceptual functions is detailed in next section and it involves coarse grain architecture represented by networked AI workstations. On the other hand, the notion of local computation supported by fine grain architectures resembling neural networks is developed in the third chapter.

Perception may be thought of as an example of a continuous problem solving operation. It is an active process during which hypotheses are formed about the surrounding environment (see 7). Sensory information acquired through vision, touch, smell, sound and proprioception is integrated to evaluate these hypotheses (8). In each of the sensory modality analog data must be first acquired and preprocessed. This stage is similar to data driven signal processing operations that are well understood in the realm of Electrical Engineering. The next stage involves segmentation and labeling of the preprocessed sensory data (2,1). And the last stage involves understanding of

the sensory information in every modality and integration for perceptual reasoning. This representational view of processing derives from generally accepted model of visual perception. Considering recent advances in computer based simulations we can implement in software any model of perception. A critical question is which mechanisms must be incorporated into hardware to guarantee human-like performance. Which architectures would make basic perceptual capabilities including learning and problem solving, feasible for autonomous mobile robots? The natural computation is based on different principles than those embodied in computers. It is a task oriented process where the current situation, including goals and drives, directly determine the next action (9). The human brain has many highly developed structures, dedicated to performing different functions, even though externally it appears to act as general-purpose system. Unravelling mysteries of perception and cognition is one of this century's major scientific challenges.

c. Neuronal architectures and parallel computation

The inspiration that AI derives from Neuroscience is based on assumption that manipulation of symbolic representations is fundamental to emergence of intelligence (2, 10). Hence, computers as symbol manipulating systems could allow us to create and test models of perception as computational activities of the brain. Since we are the keepers of information about this world we can construct the programs and data structures that internally to computer represent any concept that normally refers to external environment. The simulation running on a computer can perhaps be likened to cognitive processes that allow to reason about the consequences of physical actions before they take place. The central question is whether we could create an artificial symbolic system that uses sensory information to construct abstract representations of external world. If AI techniques will allow us to realize such symbolic behavior in a computer-based system will it have to be based on neural principles (9)? And if so how can we implement symbolic processing in terms of neural networks?

The desireability of neuronal architectures derives from massive parallelism (hence, real-time performance) and computation based on connectivity (hence, simplicity) (11, 12, 13). Parallel computation has recently become a major concern for computer science. The constraints of solid state physics limit further evolution of sequential machines to increasing speed via optical computing. And the developments in VLSI favor parallel architectures. To gain speed, one school within parallel computing paradigm assumes that computation can be performed by a pattern of connections between slow and simple processors (11, 12, 13).

Fine grain massively parallel architectures are similar to neuronal structures in the sense that they are based on millions of interacting processors. One of our immediate research problems is to investigate how can we realize such structures and how to compute with them. Because of close resemblance to anatomy of natural computing structures, this class of architectures might offer the most plausible solution to machine perception in real time (12, 14, 9, 15).

Past approaches to computer vision were based on the assumption that it can be solved in the abstract domain unrelated to the underlying physical mechanism (1, 16). Our approach differs because we constrain the problem by requiring a solution to be implementable in a 3-D connectionist architecture. The fundamental premise of *connectionism* is that individual neurons do not actively manipulate, large amounts of symbolic information (12). One of the major modes of information processing in the neural systems can be described in terms of the relative strengths of synaptic connections. Therefore, rather than using complex units that manipulate symbolic inputs, connectionist architectures computes by modulating signal with appropriately connected simple units. Hence, the computation is a form of cooperative/competitive relaxation process, taking place in a distributed net of "neural" elements.

Our approach is different from multilayer perceptrons because we propose that each unit has an S-shaped transfer characteristic (44), which can be modeled by: $V = Vmax [X / (X+k)]$, where V is the output, Vmax is the saturating level of the output signal, X is an input and the k is the input value that generates the half maximal response. This is consistent with physiological evidence for saturating membrane response and distributed synaptic inputs. inputs. The sigmoidal function allows for automatic sensitivity control, computation of relative values in context of the neighborhood and others. Thus unlike the "binary" thresholding function in perceptrons, our networks will always operate in the most optimal configuration (17).

The "neuronal" operators can have thousands of inputs and tens of outputs. A continuous output value can be generated as a thresholded hyperbolic tangent function of weighted inputs. Weights allow us to implement both positive and negative averages. Presynaptic inhibition, dendro-dendritic synapses and the concept of relative changes carrying information completes the architectural environment. These elements, allow the implementation of convergence and divergence of signal pathways as well as lateral interactions between spatially distinct nodes. Simulation of specific computing architectures is supported by UCLA-PUNNS, a neural network simulator developed in my laboratory to address the question of inseparability of function and computing substrate (18).

Principles of computation behind our simulated model are inspired by the neurophysiology of interacting neurons (19):

---o--- Concurrent computation is supported by parallel active connections between neuronal-operators, arranged in a hierarchy of layers.

---o--- Computation is performed in the analog domain and can be simulated as real-value passing networks.

---o--- For early processing stages all intra and inter layer connections are fixed and control is executed by feedback pathways which selectively modulate activity in a single operators.

---o--- Adaptive properties of the networks derive from relaxation-like behaviour, computed by each layer at the multiple scales of resolution.

---o---The cooperative and competitive modes of relaxation are computed by agonistic and antagonistic lateral interactions between neuronal operators.

---o--- Connections are modeled by weights resembling synapses with sigmoidal input-output characteristic.

---o--- Abstractions at higher levels are defined by the specific architecture of connections.

---o--- Segmentation is partially determined via bottom-up linking of many simultaneous computed images of primitive attributes.

Our principal architectural module is a three-layer computing structure (18). The INPUT layer carries a topologically correct representation of the scene. The OUTPUT layer is an abstraction which does not have to be spatially indexed to the original image. Local constraints are built into the layers. Global and local constraints are computed by the CONTEXT layer. The advantage of our concept is that it is general enough to allow the implementation of parallel architecture for signal manipulation and for aggregation of feature maps in the symbolic domain.

d. Neural net representation of perceptual knowledge.

In Computer Vision systems, programs performing visual functions are constrained by the architecture. The robustness of the human perceptual system stems from its ability to adapt/program itself. Thus novel stimuli can be processed by newly developed computing structures. Plasticity itself does not explain perception, but ability to program new knowledge and to search for alternative hypotheses is fundamental to perceptual tasks. A priori knowledge of selection criterion will always allow to exhaustively search and find an optimal model that satisfies the postulated hypothesis. The question is however, can such solution and its alternatives be identified in a reasonable time. Hence, the need for massively parallel computation in a form of neural nets.

We know that knowledge allows to optimize the search process (1). This poses a question of how to organize and represent knowledge in a memory so that it can be easily accessed at the right time (20). The factual knowledge, as opposed to "how-to" knowledge, can be organized into *networks* of associations, so that access to one part provides *connections* to other relevant parts. The knowledge about the scene must include the specifics of visually perceived objects plus the knowledge about a variety of objects in all related scenes or functions. This suggests hierarchical, as well as associational, structure. How to realise such architecture with connectionist structure, how to map the relevant knowledge onto patterns of connections and how to make it program itself by changing connectivity without "forgetting" are some of the questions that we are facing.

Perceptual knowledge must incorporate world information derived from integration of different sensory modalities. "Nihil est in intellectu quod non sit prius in sensu" (St. Thomas of Aquinas 13c), there is nothing in our intellect that did not pass through our senses. Most of our knowledge about the environment comes to us through one of the five senses. Hence, understanding the workings of these systems is a prime scientific problem. This problem is magnified in the technological realm. Vision is indispensible for autonomous mobile robots, and there is some progress in this area. Other sensory modalities are more neglected, because it is not clear how to best use them and how to implement practical solutions. In general, a solution to sensory interactions with the environment is a precursor to adaptable, intelligent performance in for example, industrial settings or in space exploration (21). The problem of best architectures or environment for studying questions related to sensory integration is open. The key questions that must be addressed are transmodal equivalences, sensory-mode specific knowledge and constraints, merging of representations specific to modality, and disambiguating conflicting modal specific information. These problems represent an important scientific challenge to implementation of machine perception.

II. MACHINE PERCEPTION LABORATORY

The coarse grain architecture of the machine perception environment consists of four networked AI workstations, each performing dedicated function (fig.1). The vision station simulates the action of the "EYE" and some higher level visual functions. The "HAND" is a separate station that provides the environment for studying manipulation and locomotion in support of perceptual task. The Ethernet fulfills the role of the spinal cord by allowing to integrate other sensory modalities, such as range, proximity, touch, etc., controlled by the "SENSE" workstation. The fourth AI workstations simulates higher level cognitive functions of the "BRAIN". The ultimate goal of this evolving architecture is to build an environment where by experimenting with global functions of machine vision and perception we could reduce scientific concepts to engineering solutions.

Although a complete theory of perception is a distant goal, both machine intelligence and humans must acquire and manipulate information from the environment. Moreover, this information must be organized into a store of knowledge that can be applied to future problems. LISP-based environments offer many advantages for experimenting with issues related to highly adaptive, multisensory based robotic systems. Such integrated environments will allow us to approach problems of vision, sensory integration, assembly and inspection as general scientific issues of planning, perception, problem solving and spatial reasoning. The machine perception laboratory (MPL) offers a realistic experimental test-bed for developing and validating various hypothesis related to robotic perception and intelligence. It also allows us to integrate and evaluate different software packages dealing with perception and manipulation.

Some software systems and tools are available currently either in academic or industrial environment, that could enhance performance and eliminate the expense of rediscovery. Of course this is feasible only if there is an environment which easily allows integration of existing systems. These packages include, among others, systems in vision, planning, decision-making, data fusion, reasoning, problem solving etc. The MPL, including all hardware/software systems, is in a continuous state of evolution and offers a diversified experimental environment spanning fields of computer science, artificial intelligence, robotics and cognitive sciences.

a. System organization

The system can be seen as a hierarchical organization of separate processes running on different workstations (22). Each dedicated station is a complete LISP-based environment extended with functions and procedures appropriate for experimenting in its domain. Part of the integration issue is addressed by extending the total environment with functions that accept, interpret and execute commands emanating from a dedicated station called "BRAIN" and send back results of their domain-specific computations. In this sense the dedicated station behaves as a lower-level entity, capable of understanding high-level commands and executing them by triggering specialized procedures appropriate to the task.

Such stations perform multitasking operations in their domains while at the same time running under the multitasking environment of the "BRAIN". This will ease the TOP-DOWN integration of the system since programming will be limited to writing specialized functions for the brain. Message-passing programming, inherent in an advanced AI environment will ease inter-task communication and the integration of new workstations. At the same time, the implementation of the system as a network of independent station will preserve their integrity, support parallel execution (vision and touch), and perhaps allow for easy integration of software written in other languages.

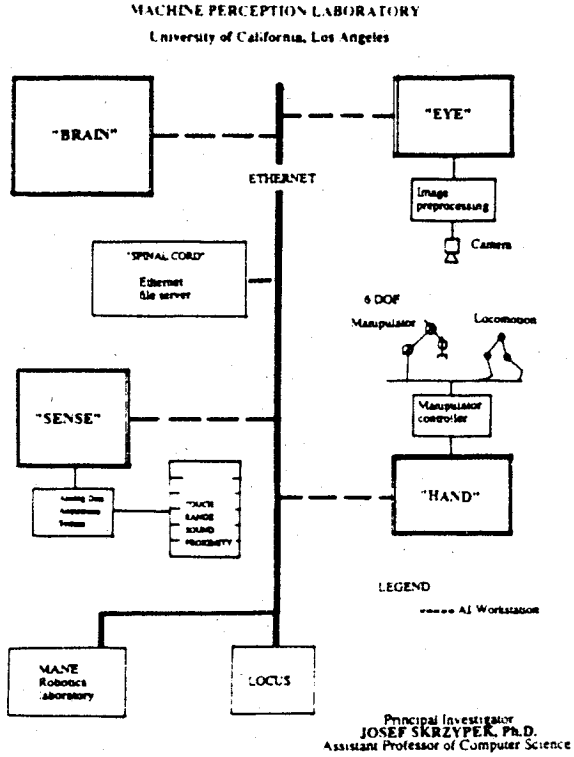


Figure 1. Global computing environment for the Machine Perception Laboratory

This implementation assumes that stations are loosely coupled. It is not our intention to study problems of close interactions between subsystems or patterns of data flow and performance in real-time. It is intended however, that the major portion of computation inherent to a specific domain, for example vision, will be performed on dedicated station called "EYE".

b. Integration

One of the key problems in setting up MPL is integration. Issues of synchronization, programmability, communication, load balancing, parallel execution are all nontrivial problems and could be analyzed by established areas of computer science. We envision that MPL will consist of a few, networked AI-based workstations and dedicated computers. Because of this, a unified LISP environment will alleviate many problems inherent in integration of such complex systems. One of the key issues of integration will be to combine symbolic and numeric computation in each sensory modality. An example of successful solution to this problem in the area of vision is given in (23).

Our initial research in the perceptual functions is focused on integration of vision with other sensory modalities. Hence, the notion of multiple networked AI-workstations, each dedicated to separate perceptual function. The LISP environment provides tools for easy integration of separate processes operating on different work stations in the network. Additionally, it allows for easier incorporation of software modules written in other languages.

The "BRAIN" plays the role of organizing problems at the task level and it assumes the responsibility of distributing computing to proper stations. Using a LISP environment to implement "BRAIN" facilitates and enhances its performance. It is relatively easy to create facilities for programming functions that can request services of remote procedures, gather high-level information from different sensory modalities, and interrupt or activate processes such as manipulation running on the other stations.

Such an environment lends itself to incremental development and testing of complex perceptual behavior. Separately developed and tested sensory or manipulation operations can be integrated as primitive functions in the "BRAIN'S" repertoire. Task-level programming, world modeling, and manipulation of symbolically represented information is fundamental to implementation of cognitive functions (24).

III. UCLA PUNNS: NEURAL NET SIMULATOR

Previous section presented an example of a coarse grain architecture, most suitable for studying global functions of perception. In this part we focus on environment for studying neural networks as physical substrate underlying local computation in perception. Physical interactions with our world demand real-time responses. If a machine is to maneuver and operate in an underconstrained, natural environment, its efficacy and survivability will also depend on how quickly it can perceive and respond (25) Natural systems solved the problem of real-time constraints by using massively parallel neural networks. The capabilities of autonomous, mobile robot are restricted by the size, weight and power requirements of the computer (26). The amount of support that a computer extracts from the machine is one of the critical factors in determining the feasibility and functional capabilities of a system. The progress in this area may come from conceptually new architectures based on neuronal principles. Hence, the need for powerful simulation tools.

Despite numerous studies over the last fifty years, we don't have a satisfactory explanation of perceptual phenomena. Part of the problem stems from inability to describe the process. Von Neumann speculated that the structure and the state of the neural network might be the simplest way to describe perception (27). Our approach to machine perception is based on assumption that the network structure yields the function and, vice versa, that the real-time function of perception implies a particular neural network structure. This approach is motivated by the reductionist view of neurophysiology where the principal notion is to explain function in terms of structure (19).

To investigate the relationship between structure and function, we have developed PUNNS (Perception Using Neural Network Simulation). PUNNS (59) is a continuously evolving environment that allows to study the functionality of massively parallel computational structures as applied to image data. The initial focus is to study neural structures that allow execution of visual functions in constant time, regardless of the size and complexity of the image. Because of complexity and cost of building a neural net machine, a flexible neural net simulator is needed to invent, study and understand the behavior of complex vision algorithms. Some of the issues involved in building a simulator are how to compactly describe the interconnectivity of the neural network, how to input image data, how to program the neural network, and how to display the results of the network.

a. Neural simulators

The theoretical properties of pseudo neural networks as applied to logical computation, learning and adaptation have been extensively explored and reviewed elsewhere (27, 28, 29, 30, 31). Many of these approaches have nothing in common with neurophysiology. Nevertheless, they do indicate the diversity of behavior that results from the interconnection of simple computational elements. PABLO is an example of a simulator that provides precise

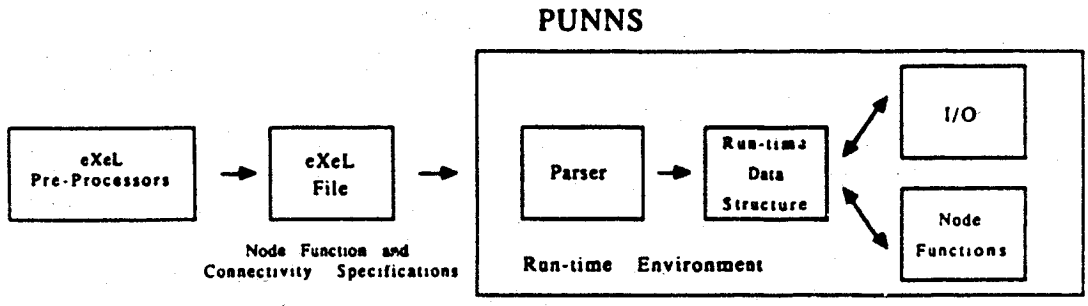


Figure 2. Block diagram of PUNNS run-time environment.

modeling of neurons and their interactions (32). Its environment closely supports many known properties of soma membrane, synaptic physiology, dendritic propagation, and axonal transmission. BOSS is another discrete-event simulator that was designed to investigate large neural networks (33). In contrast to PABLO, where each individual neuron was specified and interconnected, BOSS forms a statistical representation of the connectivity pattern. This allows for the relatively fast simulation of large connectivity patterns.

In contrast to these batch type simulators, ISCON offers the advantages of an interpreted simulator and network construction tool (34). It is written in LISP and it allows to dynamically change network connectivity and restart the simulation. The penalty for this flexibility is that large networks take prohibitively long to execute. To increase execution speed while maintaining flexibility, ISCON evolved into the Rochester Connectionist Simulator (35). RCS is a run-time environment written in C that allows user written programs to access a library of connectionist type functions, e.g. building networks, setting potentials, examining nodes.

b. PUNNS environment

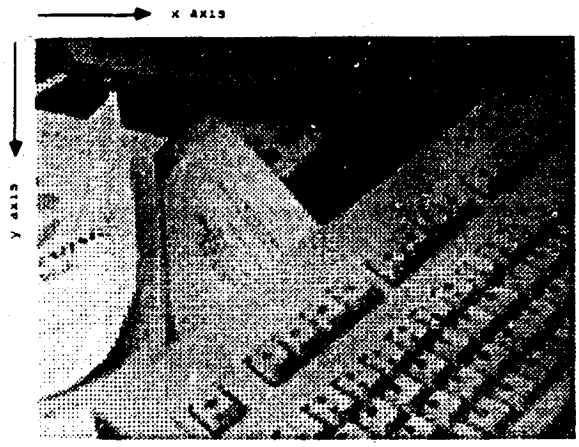
The run-time environment of PUNNS is fast and robust (fig. 2). PUNNS was implemented in C under System V and has been been ported to 4.3bsd. The underlying simulation approach used was a discrete time simulation technique that has each node visited at each simulation time step. This approach is especially useful when input data is changing every few time steps. A connectivity language (eXeL) was developed that describes the functionality of individual nodes and how they are interconnected. Complex connectivity patterns using large numbers of nodes can be generated by eXeL pre-processor routines. These are programs that output eXeL files. Hence, they are easy to modify when the connectivity pattern must be adjusted. After loading the eXeL file into PUNNS, the parser builds a data-structure which can be quickly interpreted to produce the simulation of the neural network. Changing node functions or connectivity is accomplished by reloading a modified eXeL file. Input and output to the simulation is done through graphics windows. Real images are used as a test data for the synthesized networks. A node's function can access a particular range of pixels from a graphics window and can display the result of a node, after firing, in an output window. Stimulus and response of a net can be displayed by using multiple windows. Activity levels in a layer can be viewed in one window, and the window can be saved as an image. This snapshot of activity can be then placed in an input window and newly loaded layers can continue processing from it.

In PUNNS, local connections and global mappings are used to separate the ideas of neighborhood node interactions and the connections established between functionally different blocks of nodes. Local connections are responsible for receptive field size and property, while global mappings may or may not be topologically preserving. A node's function tells what a node computes from its inputs and its temporal properties describe how the excitation level changes over time. The node is the lowest level primitive that represents an idealized, lumped parameter model of a neuron. Node description specifies inputs from other nodes input and the functions which are to act on these inputs. PUNNS also allows for dendritic input to a node, with each dendrite having a possibly unique processing function. All nodes are specified in eXeL files as follows (italics indicate a user definable parameter):

node *node-name*;
  initial-value, *length-of-history*;
  node-function;
  dendrite-1:*dendrite-function1*, *node-name16*, ... ;
  dendrite-2:*dendrite-function2*, *node-name38*, ... ;
  soma, *node-name23*, ... , *node-name42*.

The *initial-value* of the node allows a selection of different initial value. The *history-length* of a node indicates how many past excitation levels should be saved which is useful in modeling exponential decay. The *node-function* is implemented in C, it exists in the PUNNSrun-time environment and is fired when executed by the simulator. The *dendrite-function* performs the same purpose as the *node-function*. There is no provision for modeling a delay in dendritic propagation outside of synaptic transmission. The dynamic behavior of neural networks can be modulated with a *:time-delay* option that is synonymous with multiple synaptic delays.

PUNNS has been used to model and simulate pre-attentive texture segmentation (36) and the generation of matching heuristics from time-varying images (18). Figure 3 illustrates how a conceptual structure, in this case a center-surround receptive field, is analyzed using PUNNS. The structure of this receptive field, forms a strongly excitatory center and a concentric inhibitory surround. When multiple, overlapping center-surround receptive fields are applied to an image (fig. 3a.), the result is a pattern of activity that highlights discontinuities in image intensities (fig. 3b). As the transition



(b)

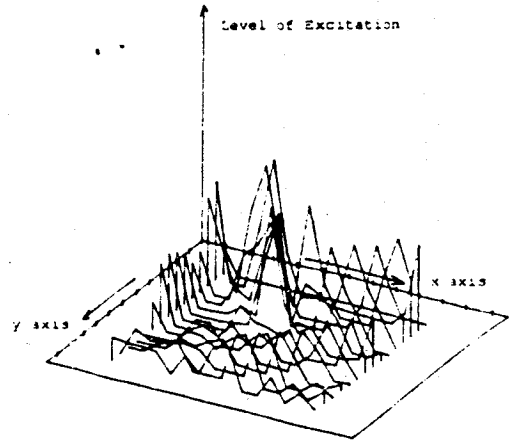


Figure 3. Example of the input image applied to the PUNNS simulating a layer of nodes with center-surround antagonistic receptive fields (a). The activities of these nodes in response to such stimulus are shown in (b)

in intensity becomes stronger, the node's excitation level increases. This structure was easily prototyped in the PUNNS environment and the simulation time was under thirty seconds.

IV. APPLICATIONS: VISION THROUGH CONNECTIONS

In this section we present examples of two early vision functions which have been implemented and analyzed using principles of neural networks.

1. Constancy preprocessor

The success of autonomous mobile robots depends on the ability to understand continuously changing scenery. Present techniques for analysis of images are not always suitable because in sequential paradigm, computation of visual functions based on absolute values of stimuli is inefficient. Important aspects of visual information are encoded in discontinuities of intensity, hence a representation in terms of relative values seems advantageous (2,3). This example deals with the computing architecture of a massively parallel vision module that optimizes the detection of relative intensity changes in space and time.

Visual information must remain constant despite the variation in the ambient light level or in the velocity of a target or a robot. Constancy can be achieved by normalizing motion and lightness scales. In both cases, basic computation involves a comparison of the center pixels with the context of surrounding values. Therefore, a similar computing architecture, composed of three functionally-different and hierarchically-arranged layers of overlapping operators, can be used for two integrated parts of the module. The first part maintains high sensitivity to spatial changes by reducing noise and normalizing the lightness scale. The result is used by the second part to maintain high sensitivity to temporal discontinuities and to compute relative motion information. Conceptually, the constraints and the rules of transformation are embedded into a computing structure which transforms the original image into two new representations. One carries the information about discontinuities in space while the other represents intensity changes in the time domain. This is consistent with the notion of space-time equivalence which suggests a hierarchical design where spatial normalization is performed before dealing with temporal domain.

Simulation results show that response of the module is proportional to contrast of the stimulus and remains constant over the whole domain of intensity. It is also proportional to velocity of motion limited to any small portion of the visual field. Uniform motion throughout the visual field results in constant response, independent of velocity. Spatial and temporal intensity changes are enhanced because computationally, the module resembles the behavior of a DOG function.

1a. Spatio-temporal considerations

Natural illumination can vary by ten logarithmic units of intensity. This exceeds the response range of artificial or biological sensors (3, 40). Hence, the first problem is how to maintain constant sensitivity to light changes over the whole intensity domain while preserving a "unique mapping" between the reflectance properties of the surfaces and perceptual notion of lightness. Linear variations of intensity usually are a consequence of nonuniform illumination (38) that can be filtered out without loosing meaningful information. The new representation of the image is expressed as relative values of intensities, that corresponds to spatial discontinuities generated by object boundaries. Absence of a DC component introduces a need for some reference point necessary to achieve lightness constancy.

Lightness constancy can be viewed as a problem of maintaining high sensitivity regardless of local or global ambient light level (39). This implies constant response when the illumination throughout the scene is multiplied by a constant. In addition, essential information such as edges must be preserved. One solution is to have sensors with a steep intensity-response (I-R) characteristic, spanning 3 log units of intensity and a mechanism that automatically shifts the operating curve to the prevailing ambient light level (40).

Nearby areas of a scene tend to have approximately equal illumination and reflectance. Hence, we use local intensity averages to set the upper and lower thresholds of the response curves. This is done automatically by adjusting the midpoints of the I-R characteristics to the local ambient light levels (40). Thereby invariance under local addition of linear illumination bias is achieved. Similar argument holds for global averages which in addition reduce sensitivity to noise by removing bias due to overall average illumination.

The detailed description of normalization is given in (17). Therefore briefly, this operation is performed by spatial operators with two antagonistic zones, center spot and surrounding annulus, better known as center/surround receptive fields (C/S-RF) (3, 41). The C/S uses lateral inhibition to emphasize contrast or relative value as novelty (42). This normalizes center signal against the spatial context information derived from the surround. Such function is equivalent to a comparison of spatially distinct areas of the image. The principle of antagonistic receptive fields is applied to all operators working on the image.

A conceptually similar problem arises in the temporal domain. Most of the objects in the real world are rigid and move with constant velocity (43). Information about them is contained in temporal discontinuities, which must be detectable regardless of ambient motion levels. Again, the limited response range of each operator necessitates continuous adjustment of operating characteristics to ambient local velocity. Hence, the system must normalize the temporal scale by resetting thresholds computed from relative, rather than absolute, values. Temporal information can be derived by comparing activities of two C/S operators of opposite polarity (3, 17). The time difference in the response waveshape of the two operators will produce a transient that carries the information about the onset/offset of change. This transient resembles a time derivative of intensity and is used to normalize the temporal scale.

It is clear from these spatial-temporal considerations that our visual system must first normalize intensity changes in space and time. And the way to subtract the DC component is to use antagonistic receptive fields implemented by lateral inhibition. The net result is a double representation of the image; one carrying spatial, and the other temporal, information. Both of them resemble the effect of convolving the original visual information with a center-surround filter resembling difference of Gaussians (DOG) (2, 3).

1b. Structural details

Our computing architecture for normalization of the lightness scale was inspired by natural vision systems (41). Major structural components include lateral interactions between neighboring elements within a layer, and converging and diverging pathways between the layers (fig. 4). Overlap between operators helps to enrich representation of the contrast information across the boundaries between different receptors. For the sake of simplicity local structures remain constant across the module.

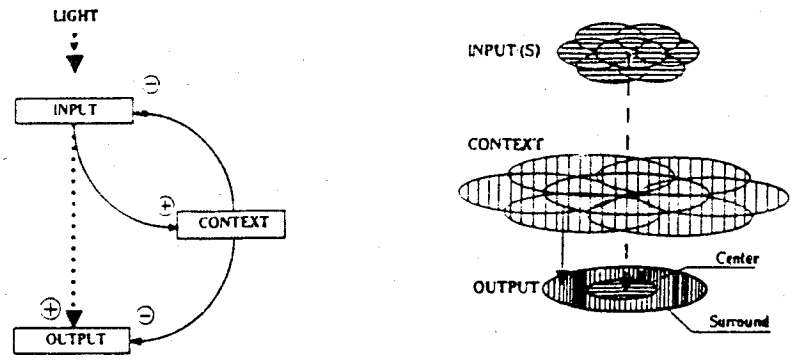


Figure 4. Flow of information in the generalized, normalization module (a). Center-surround antagonism of an output operator. Seven large context operators determine the surround response and seven smaller input operators determine the center response (b). The output operator compares the two responses.

The input to the spatial module is analogous to a layer of cone photoreceptors arranged in hexagonal array. The output operators, functionally resemble bipolar cells found in the vertebrate retina. Their responses are normalized by subtracting a local average computed by context operators. There are two types of output operators which differ in polarity and time to peak of response. The context information is always of opposite polarity to the center signal. Representing all relative values in the form of two opposite polarity masks improves stability of spatial-temporal interpolation and provides phase-like information about the original signal (45). Also, positive and negative operators differ in their coverage of the visual field and hence in spatial information. Therefore, if both positive and negative operators display a zero-crossing or a peak, it is more likely that the phenomenon is not an artifact created by noise, but a sign of significant discontinuity in intensity.

The comparison performed by output operators combines two levels of resolution in the sense that the large RF operators set the thresholds for the small ones in their area of activity. Other approaches are possible (46, 47), but for our initial implementation, we selected the simplest solution. The result computed at the output is then roughly the averaged second difference of the input intensities. Our method of combining the different levels of resolution is of particular interest because it was implemented using a simple, universal architecture based upon lateral inhibition. To simplify the simulation, we assume that the photoreceptors converging onto a given surround or output layer operator are linearly combined and that inhibition is a simple linear operation.

Fig. 5. shows combined architecture of both modules. Modularity and parallelism simplifies signal processing without any ad-hoc assumptions about image statistics. The temporal module also consists of three, functionally distinct, two-dimensional layers of C/S operators, arranged in a regular hexagonal lattice. The centers of the RF's overlap, and their sizes are different in distinct layers. To facilitate simulation, we chose to model only a small part of the visual field; hence we may assume that the sizes of the RF's remain constant throughout each layer.

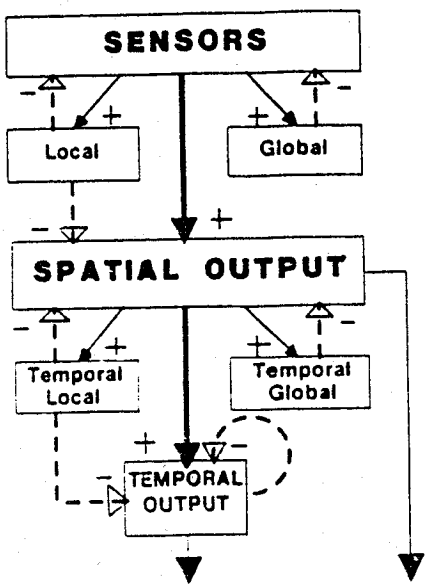


Figure 5. Hierarchical architecture of integrated spatial and temporal modules.

The input to the temporal module are two signals (I+ and I-) generated by the spatial module. They are of opposite polarity, display differences in their temporal behavior, and are regularly interspaced. Half of the temporal input operators receive I+ and the rest I-. A spatial discontinuity appearing at time t will generate a maximal response to I+ at t1 and I- at t2 with t1 and t2 not equal. This difference carries the information about the onset of temporal changes.

The time derivative is computed by an input operator which compares the information about the present input signal with values in the recent past. The source of the information about past values is feedback from the context operators. The feedback from global and local temporal context operators does not interfere with a signal normalized in space by the first submodule. This is similar to the action of local synaptic effect in amacrine cells. Context operators act to predict the future transient response to motion. The normalization of the temporal scale is achieved by shifting the velocity-response curve of the output operator over the domain of target velocities. This is based on comparison of motion in spatially seperated areas of the visual field. Conceptually, DOG of dI/dt computes temporal information which appears as a transient. However, with rapid motion, when input intensity in the center of RF fluctuates sharply, large positive and large negative derivatives could cancel during the computation of the Gaussian. Hence, the need for the DOG of the absolute value of the derivative (48). The context layer operators, which compute the feedback, rectify in the sense that negative signals are attenuated. In biological systems, the amacrine and ganglion cells rectify in order to facilitate frequency coding used in the transmission of signals over long distances (3). This rectification is functionally similar to taking absolute values in our module.

1c. Simulation results

Fig. 6. is a simple demonstration of the lightness constancy function. The output signal from the module does not change significantly as the uniform background intensity is varied. Here input, represented by the horizontal axis, is intensity on a logarithmic scale with 0 log units being equivalent to darkness. The vertical axis represents the logarithm of the difference between the extreme responses on the light and dark sides of the discontinuities.

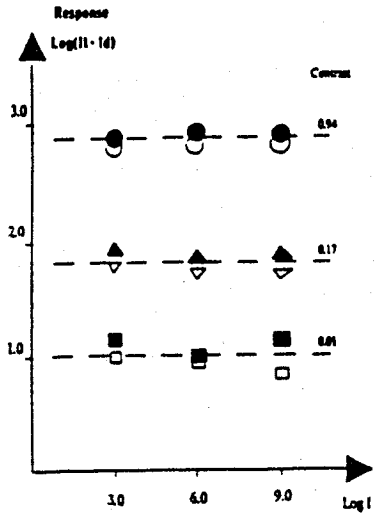


Figure 6. Response of the spatial module to increasing contrast at various levels of ambient light intensity.

The behavior of our module in response to a moving discontinuity of intensity is shown in Fig. 7. where the vertical axis represents the maximal response of an output operator in the center of the visual field. The horizontal axis represents velocity in interoperator units per iteration. One interoperator unit is the distance between two neighboring input operators. One iteration is the amount of time it takes for a signal to go from an input operator to the context layer and back to the same input operator via the immediate feedback. In all cases the input signal is a sharp discontinuity of intensity. The part of the discontinuity in the center of the visual field is moving at one velocity and the rest of it is moving at a possibly different velocity. These are called local velocity (vl) and global velocity (vg) respectively. Fig. 7. shows that if velocity is constant throughout the visual field, the response is small and almost independent of vl. However, if motion is restricted to a small part of the visual field (i.e. vg = 0) , a roughly linear response is obtained. This illustrates the fact that our module detects relative rather than absolute motion.

2. A neural net to extract motion heuristics.

The internal representation of the world that is used by a visually guided robot must be updated and maintained using the sensory data derived from the environment. Establishing a correspondence between the viewer-centered sensor data and an object-centered internal representation is an expensive computational task (49). Therefore, a roving robot must either sit for a while and contemplate it's new position, or move under assumptions which are a few steps behind the real world (50). Typically, the correspondence process forms an initial match between a perceived object and its internal model and then, as the object moves with respect to the roving robot, the orientation of the model may need to be updated to reflect current sensor information (51). This paper demonstrates how a connectionist architecture can speedup the matching of an internal 3-D model to changing edge features, by precomputing future positions of the edge features and providing the matcher with heuristic information describing in which direction to start manipulating the model.

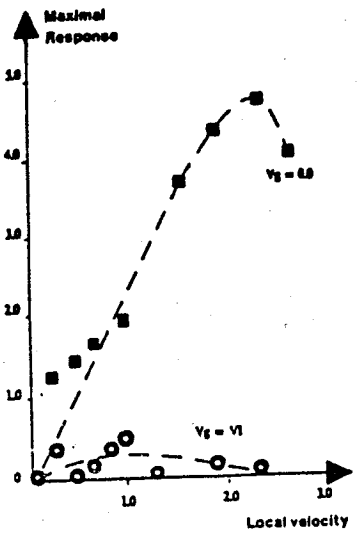


Figure 7. Motion throughout the visual field produces a small response (Vg=Vl) Motion in a small region induces a large roughly linear response (Vglob=0).

The recognition of an object must involve the matching of some input data to an internal representation of an object. The matching can be accomplished by either: 1) manipulating the data and comparing it to a set of fixed models or, 2) transforming the model to match the captured edge features. As an object in a scene moves, the 2-D projection of its boundaries and key features appear to undergo translation, rotation, and occlusion. This suggests that the second method is more natural because we do not need to compute the position of occluded edges. Also, the second method is more suitable for a goal-driven system (52). The constantly updated model becomes a representation of the world that can support scene interpretation, planning, and other higher-level cognitive functions. Manipulating the model requires the matcher to rotate and translate its internal model in an attempt to match the current edge features. In this approach the internal model is continuously trying to catch up to the real world. A speed-up would occur if the matcher received, along with the incoming data, a preliminary guess of which way the features were rotating or translating.

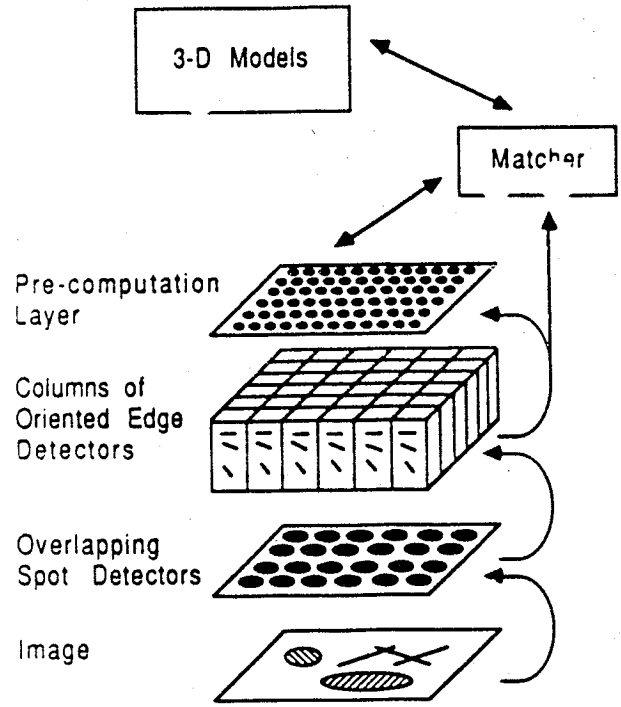


Figure 8. Overall connectionist architecture of a network used to extract motion heuristics.

Most existing matchers are based on graph theoretic algorithms which execute in exponential time with respect to complexity of the graph description (53). The matcher establishes a correspondence between the internal model representation and the edge features of an image. In this paper, we assume that this part of the matcher is given. We are concentrating on the problem of how the matcher can maintain the established correspondence as an object is undergoing smooth or discontinuous motion.

To maintain the correspondence, the matcher could precompute numerous, new orientations of the internal model and have them ready for incoming data. But this precomputation technique would be time consuming and unwieldy, since it substantially increases the graph size. Incoming data, though, can be used to give specific suggestions on how the matcher should manipulate a model. A technique for precomputing possible future positions of the edge features is the first step in formulating a model manipulation heuristic for the matcher.

By using a connectionist architecture (9), we hope to understand how visual functions can be derived from massively parallel computing structures. Additionally, neurophysiological evidence can be used to inspire possible interconnectivity solutions (17, 54,55). Our mechanism for precomputation is partially motivated by the structure of the early visual cortex which has been extensively reviewed elsewhere (56). This region of the cortex is composed of vertical slabs which contain neurons sensitive to contrast edges, of a preset orientation, that are in particular region of the visual field (56, 57). Within each slab, there is also a convergence of information related to color and motion.

We have limited our implementation of vertical slabs to the simulation of their edge orientation information. In a fashion similar to the visual cortex, edge detectors of differing orientations over the same spatial subregion are grouped together and locally interconnected. Such a group of oriented edge detectors are called a column. A column contains all of the available orientation information for its particular sub-region of the image. In the future, we hope to more realistically model the robustness of the vertical slabs in the visual cortex.

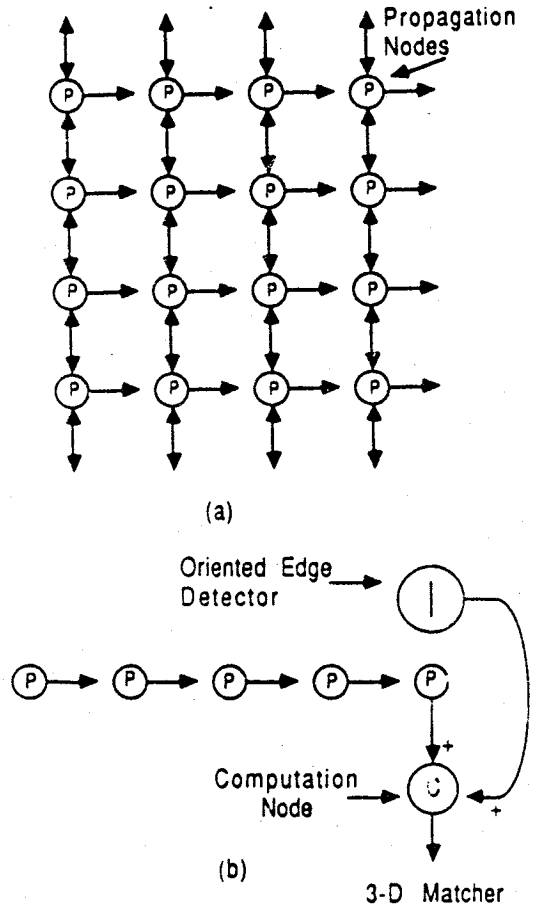


Figure 9. Propagation nodes are interconnected to propagate the direction-specific activities of edge detectors (a). A computation node requires simultaneous activities in both, its propagation node and its edge detector, before it will signal the response.

Fig. 8 outlines the computational hierarchy of the architecture. The light from a scene is initially transduced into electrical signals by a layer of photosensors. These signals are then processed by spot detectors which are sensitive to local changes in image intensity. The center receptive fields of the spot detectors overlap each other by thirty percent. The output from the spot detectors are grouped to form oriented edge detectors which are then organized into columns. It should be noted that this implementation deliberately differs from the known neurophysiological data because of the limitations of our simulation tools. Surrounding each edge detector are propagation nodes which compute where the edge may move in the future by exciting the propagation nodes in adjacent columns. Oriented edge information is used by both the precomputation layer and the matcher. The precomputation layer gives the matcher heuristic information on the direction of a moving edge feature. Computation nodes which are in this layer are able to guess at the direction of an edge by comparing the excitation levels of oriented edge operators and the surrounding propagation nodes.

The lowest layer of the architecture extracts changes in image intensity by using center-surround receptive fields has been detailed in (18). Briefly, the image is first filtered by a layer of nodes with center/surround antagonistic receptive fields. To reduce the simulation complexity, this layer was modeled using a convolution operator.

The analyses of the information available from motion makes it apparent that there are only few possible directions that edge feature could take without violating the heuristics used for matching points in separate images (58). Considering only rigid physical objects with limited velocity, the motion is limited to a few possible next-frame positions and directions. Hence in principle it is possible to simultaneously tell the matcher where the edges are and how they are moving.

To accomplish this objective, we organize the oriented edge detectors within a sub-region of the image into a column and then bring the columns together to form a cube. A transverse slice of the cube contains all of the edge detectors of a particular orientation over the entire image. When an edge becomes active, indicating that the current image has an edge feature at that location and orientation, we want to use that fact to prepare for future movement of that edge feature.

A moving edge feature can at most activate one of six, nearest-neighbor edge detectors in our hypercolumn. To monitor this change, each oriented edge detector in a column is connected to six propagation nodes (p-nodes), four translational and two rotational. Thus, a specific p-node will transmit the activity of its edge detector in one of the six possible directions. By propagating the excitation of an edge detector, the p-nodes prime the network for specific, future orientations of an edge feature (fig. 9).

A computation node (c-node) combines the information from an oriented edge detector and its associated p-nodes. A c-node will only fire when its edge detector and one p-node are high. Of course prior to arrival of the edge feature, high activity of one p-node implies potential direction of motion that can be signaled to the matcher.

2a. Example

Fig. 10 illustrates the changing excitation levels of the p-nodes and c-nodes over time. In this example, a bar is moving from left to right across the visual field (Fig. 10a) Fig. 10b demonstrates how the excitation levels of edge detectors are being propagated, in a rightward direction, by the +y translational p-nodes. When both the p-nodes and the edge detectors are excited, the c-nodes will momentarily fire (fig. 10c) and provide heuristic information to the matcher.

The precomputation layer of our connectionist architecture can provide heuristic information useful in matching 3-D models to time-varying edge features. If the velocity of an edge feature should exceed the propagation rate of the p-nodes, then the c-nodes will not be excited and the matcher will not receive any heuristic information. The matcher could interpret such an edge as being part of either, a new object in the scene or, an object that is undergoing discontinuous jumps.

CONCLUSION

New approaches to machine sensing and perception were presented. The motivation for crossdisciplinary studies of perception in terms of AI and Neurosciences is suggested. The question of computing architecture granularity as related to global/local computation underlying perceptual function is considered and examples of two environments are given. Finally, the examples of using one of the environments, UCLA PUNNS, to study neural architectures for visual function are presented.

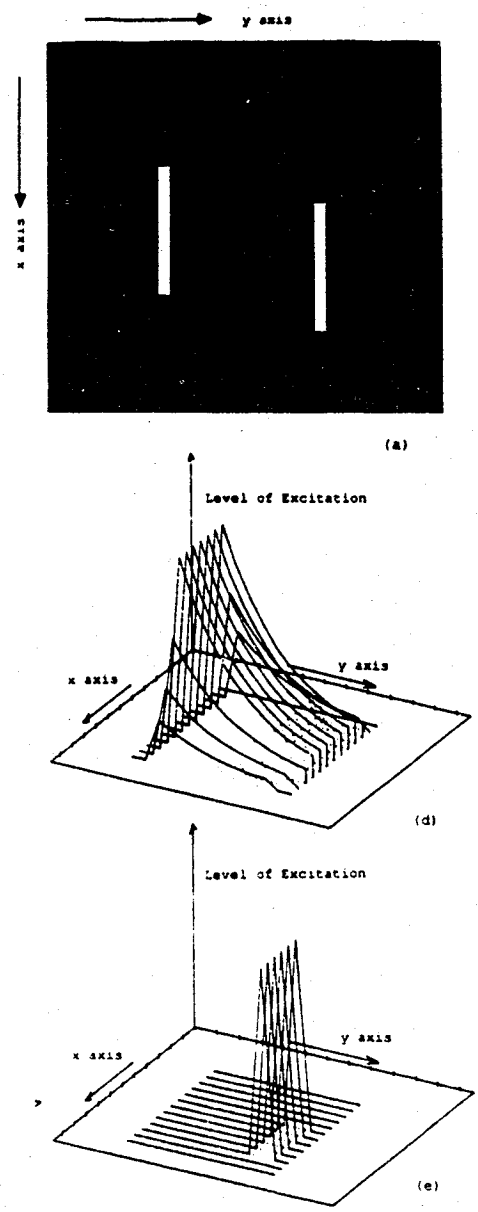


Figure 10. The stimulus is a time-varying image of the vertical bar is moving from left to right (a). The P-nodes propagate the exponentially decaying signal about vertically oriented edge moving in the +Y direction (b). When the bar moves to the right, the C-node becomes active and sends the information to the matcher that this edge feature has undergone left to right translation

REFERENCES

1. Barr, A., & Feigenbaum, E., The handbook of artificial intelligence. 1982, Heuristech Press.
2. Marr, D., Vision. 1982, Freeman and Co.
3. Rodieck, R.W., *The Vertebrate Retina*, W.H. Freeman and Co., San Francisco, 1973.
4. Barlow, H.B., Why have multiple cortical areas? Vis. Res. 26 pp.81-90, 1986.
5. MacLeod.R.B & H.L.Pick Jr., Perception. 1974 Cornell University Press.
6. Schank, R., & Abelson, R., Scripts, Plans, Goals and
7. Rock.I., The logic of perception, 1983, MIT Press.
8. Henderson T.C., Wu So Fai, and Hansen C., "MKS: A Multisensor Kernel System," IEEE Trans. Sys., Man, Cybern., vol. SMC-14, no. 5, pp. 784-791 (1984);
9. Feldman, J.A., "Connectionist Models and Parallelism in High Level Vision", *Computer Vision, Graphics, and Image Processing*, 31, pp. 178-200, Feb. 1985.
10. Hinton,H., & J.Anderson, Parallel models of associative memory, 1981 L. Earlbaum Associates Inc.
11. Rosenblatt, F., *Principles of Neurodynamics*, Spartan Books, Washington D.C., 1962.
12. Feldman, J.A., and Ballard, D.H., "Connectionist Models and Their Properties", *Cognitive Science*, 6, pp. 205-254, 1982.
13. Hopfield, J.J., "Neural networks and physical systems with emergent collective computational properties." Proc. Natl. Acad. of USA 1982, 79, 2554-2558.
14. Ballard, D.H., "Parmeter Networks: Towards a Theory of Low Level Vision", TR 75, Computer Science Dept., University of Rochester, April 1981.
15. Hopfield, J.J. & Tank, D.W., "Neural computationa in constraint satisfaction problems and the traveling salesman." Biol. Cyber, in press.
16. Winston P.H., "Artificial Intelligence," Addison-Wesley, Reading, Mass. (1984).
17. Skrzypek, J., "A unified computational architecture for preprocessing visual information in space and time." ANRT/SPIE Proc of Intl. Symp. on Computer Vision and Intelligent Robots, Canes, France 1985.
18. Gungner, D., and Skrzypek, J., Connectionist Architecture for Matching 3D Models to Moving Edge Features, *SPIE Conference on Intelligent Robots and Computer Vision*, Vol. 726, October, 1986.
19. Kuffler, S.W., and Nichols, J.G., *From Neuron to Brain*, Sinauer Associates, Inc. Massachusetts, 1976.
20. Buchanan,B.G., & Shortliffe, E.H.,(eds) Rule-based Expert Systems: The mycin experiment of the Stanford Heuristic Programming Project, Addison-Wesley, 1984.
21. Brady,M., & Paul,R., eds. Robotics Research, 1984, MIT Press.
22. Feldman, J., et al., "The Stanford artificial intelligence project", Proc. IJCAI, Washington D.C., 1969
23. Stiber, M., & Skrzypek, J., "LISP based PC Vision Workstation." SPIE Conf. Image Understanding and Man Machine Interfaces. Los Angeles. 1987
24. Michalski, R.S., Carbonell, J.G., & Mithcell, T.M., (eds.) Machine learning: An Artificial Intelligence Approach, Tioga publishing Co. Palo Alto, 1983.
25. Thagard, P., Parallel Computation and the Mind-Body Problem, *Cognitive Science*, 10, 301-318, 1986.
26. Toffoli, T., Physics and Computation, *International Journal of Theoretical Physics*, Vol. 21, Nos. 3/4, 1982.
27. Von Neumann, J., Second lecture of Theory and Organization of Complicated Automata, in *Theory of Self-Reproducing Automata*, Burks, A.W. ed., University of Illinois Press, 1966.
28. McCulloch, W.S., and Pitts,W., A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, Vol. 5, 1943.
29. Hebb, D.O., *The Organization of Behavior*, John Wiley, New York, 1949.
30. Rosenblatt, F., *Principles of Neurodynamics*, Spartan Books, Washington D.C., 1962.
31. Rumelhart, D.E., McClelland, J.L., et.al., *Parallel Distributed Processing*, MIT Press, 1986.
32. Perkel, D.H., A Computer Program for Simulating a Network of Interacting Neurons, *Computers and Biomedical Research*, Vol. 9, pp. 31-43, 1976.
33. Wittie, L.D., Large-scale Simulation of Brain Cortices, *Simulation*, Vol. 31, Num. 3, September, 1978.
34. Small, S.L., Shastri, L., Brucks, M.L., Kaufman, S.G., Cottrell, G.W., Addanki, S., ISCON: A Network Construction Aid and Simulator for Connectionist Models, *University of Rochester, Department of Computer Science Technical Report*, TR 109, April, 1983.
35. Fanty, M., Goddard, N., User's Manual Rochester Connectionist Simulator - Draft, *University of Rochester, Department of Computer Science Technical Report*, September, 1986.
36. Mesrobian, E., and Skrzypek, J., Connectionist Architecture for Computing Textural Segmentation, *SPIE Conference on Image Understanding and the Man-Machine Interface*, Vol. 758, January, 1987.
37. Hubel, D.H., and Wiesel, T.N., Brain Mechanisms of Vision, *The Mind's Eye*, W.H. Freeman and Company, New York, 1986.
38. T. Binford., "Inferring Surface Information from Images", Artificial Intelligence volume 17, pp. 205 -241, 1981.
39. E.Land. and J.J.McCann., "Lightness Theory", Journal of the Optical Society, volume 61, number 1, January 1971, pp. 1 -11.
40. J. Skrzypek and D. Shulman, "Preprocessing Using Multiresolution Lateral Inhibition", SPIE Proc. Cambridge Conf. on Intelligent Robots and Computer Vision, Cambridge 1984, 91-97.
41. J, Skrzypek. and F, S. Werblin., "Lateral Interactions in the Absence of Feedback to Cones", Journal of Neurophysiology volume 49, pp.1007-1016. 1983.
42. F, Ratliff., Mach Bands: Quantitative Studies on Neural Networks in Retina, Holden Day 1965.
43. E. C. Hildreth, "The Measurement of Visual Motion", Ph.D. Thesis M.I.T., Department of Electrical Engineering. 1983.
44. K, Naka. and Rushton. W.A.H., "S Potentials from Intensity Units in the Retina of Fish (Cyprindidae)", Journal of Physiology, volume 185, pp. 587-599. 1966.
45. S. W. Zucker and R. A. Hummel, "Receptive Fields and The Representation of Visual Information", IEEE Conference On Pattern Recognition, Montreal 1984, 515-517.
46. J.F. Canny, "Finding Edges and Lines in Images", M.I.T. Artificial Intelligence Laboratory Technical Report 720. 1983.
47. D. Terzopoulos, "Multilevel Reconstruction of Visual Surfaces: Variational Principle and Finite Element Representations " in Rosenfeld, A., (ed.) Multiresolution Image Processing and Analysis, Springer Verlag, 1983.
48. D.Shulman and J.Skrzypek, "Multiresolution temporal preprocessing and lateral inhibition.", SPIE proc. Cambridge Conf on Intelligent Robots and Computer Vision, Cambridge 1985.
49. Shepard, R.N., and Cooper, L.A., *Mental Images and Their Transformations*, MIT Press, 1986.
50. Lowrie, J.W., Thomas, M., Gremban, K., and Turk, M., The Autonomous Land Vehicle (ALV) Preliminary Road-following Demonstration, *Intelligent Robots and Computer Vision*, SPIE Vol. 579, pp. 336-350, Cambridge, Massachusetts, 1985.
51. Tsuji, S., Osada, M., and Yachida, M., Tracking and Segmentation of Moving Objects in Dynamic Line Images, *IEEE Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 6, November 1980.
52. Pinker, S., Visual Cognition: An Introduction, in *Visual Cognition*, Pinker, S. ed., pp. 1-63, MIT Press, 1985.
53. Ballard, D.H., and Brown, C.M., *Computer Vision*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982.
54. Marr, D., and Poggio, T., A Theory of Human Stereo Vision, *Proc. Roy. Soc. London B*, Vol. 204, pp. 301-328, 1979.
55. Ballard, D.H., Parameter Nets, *Artificial Intelligence*, Vol 22, pp. 235-267, 1984.
56. Hubel, D.H., Evolution of Ideas on the Primary Visual Cortex, 1955-1978: A Biased Historical Account, Nobel Lecture, December 8, 1981, *Les Prix Nobel*, Stockholm, Sweden, 1981.
57. Marr, D., and Hildreth, E., Theory of Edge Detection, *Proc. R. Soc. London B*, Vol. 207, pp. 187-217, 1980.
58. Prager, J.M., Segmentation of Static and Dynamic Scenes, *COINS Technical Report*, 79-7, Computer and Information Science, University of Massachusetts, May 1979.
59. Gungner, D., & Skrzypek, J. "UCLA PUNNS - A neural network machine for computer vision." SPIE Conf. Image Understanding adn Man-Machine Interfaces., Los Angeles, 1987.