NAVY Topic 113: Object Recognition Chip

A high-speed object recognition chip based on a biologically-realistic Hybrid Temporal Processing Element



Progress Report # 2

Intelligent Reasoning Systems, Inc.
P.O. Box 30001, Dept. 3ARP
Las Cruces, NM 88003



Wednesday 4 August 1993

This report traces the progress made by Intelligent Reasoning Systems, Inc. (IRSI) on contract N00014-93-C-0118, from June 1, 1993 to July 31, 1993. The structure of this report is as follows: introduction and Phase I objectives, progress to date, projected progress, and appendices containing example programs.

**Technical Objectives:**

The technical objective of this Phase I effort is to evaluate the feasibility of a binocular object recognition system for the INtegrated Active VISION System (INAVISIONS) currently under development at IRSI. The INAVISIONS is based on a custom Very Large Scale Integration (VLSI) Hybrid Temporal Processing Element (HTPE) developed by IRSI (patent pending) [1-4].

Active vision implies that the image collection and processing operations form a single, integrated, adaptive system in which high-level processes, such as object recognition, can augment lower-level processes through feedback connections and can itself be controlled by higher levels. For example, the object recognition process can instruct the oscillating saccadic perception system to enhance the image resolution by a particular amount to assist in the object recognition process. Higher level perception systems can guide the object recognition processes by using expectations of what objects might be present in the image. Therefore, the INAVISIONS behaves a single image collection and processing system with coupled low- and high-level capabilities.

The IRSI INAVISIONS incorporates principles derived from experimental analysis of mammalian visual systems. Mammalian visual systems employ multiple concurrent, cross-coupled processing pathways to simultaneously infer location, motion, shape, color, and texture from images of an object obtained asynchronously and from multiple points of view (e.g. [5-6]). Processing occurs asynchronously, with continuous feedback between early and later processes. For example, low-resolution location and motion detection guide eye movement and visual attention through a fast feedback loop, while also contributing to slower, higher-resolution downstream motion, shape, and texture processing.

The INAVISIONS design is based on an asynchronous analog encoding of location and motion data, and the custom VLSI HTPE. The HTPE closely models the generic spiking neurons of the human brain [1-4]. Spiking neurons perform time-based processing using impulse-like Action Potentials (APs) and slowly time-varying Post Synaptic Potentials (PSPs). HTPEs can be used to encode and process information in terms of amplitude, frequency, phase, and time delay. In HTPE systems, transient and steady-state waveforms are used to provide real-time processing capabilities. The HTPE can operate on analog data at frequencies in excess of 100 MHz, allowing rapid oversampling methods to be used for resolution enhancement, motion detection, and multiple template matching. HTPEs have low device count and power dissipation, and can be fabricated in small layout areas. The INAVISIONS is intended for eventual on-board application in robots,

intelligent machine tools, and other autonomous sensory-motor systems that require visible, IR, UV, or similar input. The basic architecture developed for the INAVISIONS will, however, also be useful for any sensory input that can be represented in a plane, and for which spatial continuity is a relevant constraint.

The initial stage of the INAVISIONS collects data asynchronously from the primary sensor, and provides a parallel analog encoding. This encoding allows preprocessing circuitry for tasks such as low-level resolution enhancement and motion-detection to be built into the retina, and to operate on data from each pixel or neighborhood of pixels in parallel. The preprocessed data is then encoded in asynchronous pulse streams and mapped, again in parallel, to hierarchical feature detection systems in which spatial relations between processing elements corresponding to pixels or pixel neighborhoods are preserved. The processing layers in this initial INAVISIONS stage correspond to the early processing stages in the retina of the Human Visual System (HVS). The initial stage of the INVASIONS, from here on referred to as the INtegrated Saccadic Perception Image Resolution Enhancement System (INSPIRES), was developed under an Army Phase I SBIR effort and is currently in the Phase II review process.

Successful development of the object recognition chip depends on the development of the low-level feature extraction systems of the INAVISIONS. The preliminary development of these systems was performed under the Army INSPIRES project and will be further developed over the course of this project. Since the feature extraction systems provide direct input to the binocular object recognition system under development in this Phase I effort, their concurrent design will allow them to be tailored so as to best compliment each other.

The specific objectives of this Phase I effort are:

1. The INAVISIONS primary feature detectors will process data from each pixel neighborhood in parallel. Fusing data from two primary feature detectors raises two hardware issues: a) data will probably have to be multiplexed off of the feature-detector chips and on to the fusion chip; b) multiple fusion chips may be needed, raising the issue of preserving spatial coherence between overlapping fields. Alternative strategies for addressing both of these issues need to be evaluated.

2. The primary depth cue is binocular disparity. Disparity between the primary feature maps is sensitively dependent on the resolutions of both the retinas themselves and the downstream processing. The mapping of disparity to depth for various realistic combinations of retinal spacing, retinal resolution, and feature resolution needs to be investigated.

3. The binocular fusion circuitry for calculating 3D location and motion of objects and features needs to be designed and implemented. An appropriate encoding for combined object and motion data needs to be developed. The fusion system design needs to be general enough to work for a variety of depth-to-disparity mappings.

4. Analog logic needs to be developed for combining primary features into simple shapes as the first step in shape recognition. Conventional Artificial Neural Networks (ANNs) can classify observed shapes into learned classes, but require high connectivity if many classes are used. A hierarchical classification system for constructing complex shapes from simple ones may be more practical than a conventional ANN for hardware implementation. The relevant tradeoffs between these two approaches need to be identified and evaluated.

5. Occlusion is an excellent depth cue, but can only be used if occluding and occluded objects can be distinguished. The ability of the shape-classification system for distinguishing occluded objects needs to be tested.

6. In general, the optimal split between object recognition for attention and targeting and shape recognition needs to be identified, as do the points at which feedback from higher-level processing can be used to disambiguate objects and shapes.

**Progress to Date:**

SUMMARY OF FIRST WORK PERIOD

In the first work period, the overall structure of the Feature and Object Recognition SystEm (FORSE) was designed. The FORSE is intended for use in the complete INAVISIONS. The INAVISIONS, and hence, the FORSE are fully-parallel systems that are designed to be implemented in state-of-the-art three-dimensional VLSI circuitry. The structure of both systems will be examined in the following section on the projected progress for the final work period.

Conventional Cellular Neural Networks (CNNs) were examined in the first work period for use as spatial feature extraction and image transformation systems. CNNs require only spatially homogeneous, local connections and are amenable to analog VLSI implementation. The network structure is two-dimensional and maintains a one-to-one mapping from image pixels to network processors (referred to as cells). The strength and pattern of the local connections determine the dynamics of the overall network. CNNs may perform high-speed, spatial image processing operations such as line and edge detection, region filling, connected-object detection, and primitive handwritten character recognition (as determined by the local connections). A software tool was developed by IRSI for the design and simulation of large scale CNNs and example simulations were provided (vertical edge detector, horizontal edge detector, concave and convex edge detector, and hole filler). The simulator allows a sequence of local connections with different effects on the network dynamics to be applied in series. These sequences effectively forming CNN programs.

Conventional CNNs are time-independent processors which limits their application to spatial image processing and simple temporal processing. For temporal applications, a complex sequence of static spatial operations are performed over a series of image frames. A number of operations must be performed on each frame just to estimate one-dimensional object motion in a simple image containing only one object. If multiple objects are involved or velocity and acceleration information is to be determined, the time-independent spatial processing of conventional CNNs will prove to be inadequate.

Conventional CNNs find their motivation in the *simple* and *complex* cells of the HVS [7]. Simple cells which are sensitive to lines and edges at particular orientations and at all locations across the visual field are analogous to conventional CNNs in that they perform spatial processing only. Complex cells, on the other hand, are sensitive to lines and edges at particular orientations moving in certain directions. Stationary features do not evoke a response from complex cells. HTPE-based simple and complex cell networks have been developed which provide the functionality of their biological counterparts but are also capable of more complicated operations. The simulated behavior of these networks is presented in the second work period section.

PROGRESS IN SECOND WORK PERIOD

Data Multiplexing and Demultiplexing

Due to the fully parallel architecture adopted for the INAVISIONS and FORSE, multiplexing methods for the transport of data off one chip and onto another are not a major issue at this point. Since subsequent processing stages in the INAVISIONS may require data multiplexing, however, we have developed several methods for encoding multiple pieces of data onto a single asynchronous pulse stream. Methods for extracting the individual pieces of data from the pulse

stream have also been developed. The location of the data piece in the pulse stream is used to maintain spatial coherence.

The primary technique for the transmission and storage of information in HTPE networks is to encode the information in asynchronous pulse steams. There are multiple methods for encoding information in pulse streams, including frequency, phase, and pulse width modulation. Combinations of these methods and both linear and non-linear mappings are possible. A common feature of all of these data encoding techniques is that the data is encoded on the temporal characteristics of the waveform. Time-based, asynchronous data encoding provides several distinct advantages: first, encoding of analog values on the time axis provides immunity to signal noise, which plagues conventional analog systems; second, due to the fine temporal sensitivity of the HTPE, a large amount of information (as compared to conventional digital encodings) can be encoded in short time delays between two APs; and third, if priority or frequently used information is encoded as short time delays it will automatically be processed at a higher rate due to the asynchronous nature of HTPE processing. Figure 1 illustrates a number of time-based, pulse-stream encoding methods. Figures 1a, 1b, and 1c show frequency, phase, and pulse-width modulated pulse streams, respectively. Figure 1d shows a combination of frequency and pulse-width modulation, and Fig. 1e shows the most efficient and commonly used technique which combines time-delay and pulse-width modulation.

The frequency, phase, and pulse-width modulation techniques encode a single analog parameter in the average frequency or pulse-width of a single pulse stream or the average phase shift between two equal frequency pulse streams. The technique that combines frequency and pulse-width modulation can store two analog parameters in a single pulse stream. A disadvantage of the phase encoding method is that it requires two pulse streams or a common global reference pulse stream to encode a single piece of information. Time-delay encoding can be used to store N parameters and M control flags in a single pulse stream. The N data parameters are stored as N time-delays (inter-pulse spacings i.e. time between pulses) and the M control flags are stored as M pulse-widths of selected pulses throughout the stream. For example, the pulse stream of Fig. 1e contains six parameters stored in the six inter-pulse spacings of the periodic stream (i.e. the inter-pulse spacings repeat every six spacing, and therefore, the period of the stream is the sum of the six inter-pulse spacings). Note that to have N inter-pulse spacings requires N+1 pulses. Since the streams are periodic, the last pulse in an oscillation serves as the first pulse in the next oscillation.

Two different references can be used to quantify the inter-pulse spacing each with different inter-pulse-spacing/pulse-width dependencies: 1) if the inter-pulse spacing is measured from the rising-edge of a pulse to the rising-edge of the next pulse in the series, the minimum inter-pulse spacing must be greater than the maximum pulse width; and 2) if the inter-pulse spacing is measured from the falling-edge of a pulse to the rising-edge of the next pulse in the stream, the inter-pulse spacing and pulse width are independent.

Another method of time-delay encoding in which a set of inter-pulse spacings or pulse widths are used to encode a single variable allows more information to be encoded in a shorter amount of time, and therefore, increases the encoding density. This is illustrated in Fig. 2, which shows a single parameter being encoded by the two different methods of time-delay encoding. If both methods assume a one time-unit resolution, the first method (top) requires up to 1001 time units to store one of 1000 values (1000 distinct inter-pulse spacings), and the second method (center) requires up to 33 time units (one time unit is required for the pulse width of the first three pulses). The first method employs a straight linear mapping from the variable to the time axis, while the second method employs a non-linear base-ten encoding over three inter-pulse spacings ($10^3$) to encode one of 1000 values. The second method requires less time to encode the information but requires more complex decoding techniques.

In general, non-linear time encoding methods can greatly increase the system operation speed due to the time-based nature of the FORSE and HTPE networks. If the encoding and decoding circuitry, however, becomes too complex, the time savings may be lost in the encoding/decoding process itself. Complex circuitry may also require large amounts of expensive VLSI real estate which is a concern in any large-scale integrated system. Information encoded using both linear and non-linear methods such as the base-ten technique are directly processable by the same HTPE networks. The primary consideration is, therefore, which method is the least expensive in terms of time and VLSI area. The exact encoding methods to be utilized will be determined during the Phase II effort but will definitely be based on the N frequency, M pulse-width modulation technique.

The mapping of a particular analog value into HTPE temporal parameters can either be linear or non-linear and independent of the encoding method, due to the non-linear dynamics of the HTPE circuitry itself. To encode an analog value in a pulse stream, it is typically used to bias an HTPE network. This bias causes a particular temporal behavior (i.e. frequency or pulse width). Figure 3 shows the non-linear mapping between the HTPE bias voltages and the duration of the temporal parameter which they control (i.e. AP inter-pulse spacing and pulse widths or PSP delays or durations are all controlled by DC biases). These non-linearities result from the Voltage-Current characteristics of the NMOS (rightmost curve) and PMOS (leftmost curve) devices that make up the HTPE circuitry. The bias voltages of the current HTPE design range from zero to five volts. Examination of Fig. 3 reveals the overall non-linearity is composed of two linear regions (one with a small slope and another with a large slope) connected by a non-linear ($x^2$) region. Linear mappings can be produced by constraining the bias values to the linear portions of the HTPE non-linearity or by determining the proper non-linear mapping of the analog value to be encoded. A proper non-linear mapping would transform the value to be encoded into a bias voltage that will produce a linear time parameter encoding. For example, if a particular analog value is to be encoded as an inter-pulse spacing, it will be non-linearly transformed into a bias voltage that when applied to the HTPE network will cause a linear mapping of the analog value to inter-pulse spacing.

Figure 4 shows the HTPE oscillator structure that generates N frequency, M pulse-width pulse streams (referred to as an FPWO). A single "start" pulse causes the encoded oscillation to begin. Each axon hillock in the outer loop fires an excitatory synapse input on the axon hillock next in line, which causes the outer loop to oscillate. Each excitatory synapse is also connected to a common axon hillock which fires a pulse every time an outer loop axon hillock fires a pulse (the common axon hillock "hC" is acting as a pulse OR-gate). The time-delay of the synapses determines the inter-pulse spacings of the pulse stream produced by the common axon hillock. To encode information in the inter-pulse spacings of the stream, the data is used to determine the bias voltage that is applied to the excitatory synapse to control their time-delay. For example, if a position sensor output voltage or current is to be encoded as an inter-pulse spacing, it will be converted to a voltage (by simple resistive networks) with a range that is compatible with the voltage range of the MOSFET devices on which the HTPE is based. The converted voltage is directly applied to the excitatory synapses of the oscillator.

To encode control actions in the pulse widths of selected pulses, the control actions are used to set the bias voltage (also using simple resistive networks) that determines the width of the pulses generated by the common axon hillock. One possible control action to be encoded is the location of the first pulse in the oscillation which is required to extract the information contained in the stream in the proper order. A simple way to encode this control function is to make the first pulse in the cycle double the width of the other pulses. Other control actions can similarly be encoded in the pulse widths of other pulse in the stream.

Figure 5 shows the simulated behavior of a linearly-encoded FPWO. For the particular oscillator shown N=6 and M=1. The top plot of the figure shows an oscillation with inter-pulse spacings of 5, 10, 15, 20, 25, and 30 time units. The first pulse of the oscillation is two time units wide while the other pulse are a single time unit wide. The bottom plot of the figure shows a similar pulse stream with inter-pulse spacings of 5, 10, 20, 30, 15, and 20 time units. For simplicity, we will refer to the pulse streams generated by the FPWO as NM pulse streams.

The individual pieces of data encoded in an NM pulse stream can be extracted by an HTPE Serial-to-Parallel Converter (SPC). The converter, shown in Fig. 6, transforms an NM serial AP stream into N+1 parallel AP streams of equal frequency, with relative phase shifts equal to the N respective subperiods. For example, if the inter-AP spacings of a particular three-frequency AP stream are 12, 22, and 7 time units; the four parallel output streams produced by the SPC will have equal periods of 41 time units, with a 12 time-unit phase shift between streams one and two, a 22 time units-phase shift between streams two and three, and a 7 time-unit phase shift between streams three and four. The actual period of the AP streams must also include the pulse widths of the pulses contained in one period of the stream (i.e. if each of the three APs that define one period of the stream are one time unit wide, the actual period of the 12-22-7 stream is 44 rather than 41 time units). The period of the output AP streams is therefore, a function of both inter-pulse spacings and pulse widths. The phase shift between the streams, however, is a function of only the inter-pulse spacings. The SPC is an edge-triggered device that responds only to the rising edge of the APs in the input stream but does not process or notice the information contained in the pulse widths of the APs in the stream. Additional pulse-width sensitive networks extract the control actions encoded in the AP durations before the stream is input to the SPC. For example, the unity-width pulse used to synchronize an NM pulse stream for data extraction can be detected by a pulse-width sensitive network that resets and fires the SPC when the unity-width pulse is detected in the stream. Other control actions such as SPC output data routing can similarly be encoded and extracted.

The SPC operates as follows. The N-frequency stream is applied to the N+1 input excitatory synapses, labeled "e1" through "eN+1". The synapse e1 is different from the other input synapses in that it has a high enough amplitude to fire "h1" by itself. When the first pulse in the input stream arrives, it causes e1 to fire h1 (the other input synapses are also fired by the input pulses, but are not sufficient to fire their corresponding hillocks by themselves). When h1 fires a pulse, it is fed back to h2 through synapse "ef1". Synapse ef1 does not have a high enough amplitude to fire h2 alone, but persists long enough that when the next input pulse arrives and the input synapses are fired, the amplitude of e2 plus that of ef1 are sufficient to fire h2. When h1 initially fires, it also feeds back to its own input through inhibitory synapse "i1". This inhibitory input prevents h1 from firing by reducing its net input below threshold. When each hillock (h2 through hN) fires, it resets the feedback synapse connected to its input, and therefore, will not fire again until its feedback synapse is refired by the hillock preceding it. This process proceeds until the Nth hillock is reached, which when it fires resets the inhibitory connection on h1. Now when the next input pulse arrives, both h1 and hN+1 fire and the whole process repeats.

Figure 7 shows the simulated behavior of the SPC. A three-frequency input stream (represented by "h4", top) is converted into four parallel output streams (represented by "h20", "h21", "h22", and "h23", center). The inter-pulse spacings of the input stream are 12, 22, and 7 time units and the phase shifts between the output pulse streams are 12, 22, and 7 time units. The fourth output stream (h23) occurs simultaneous to the h20 stream and due to the line patterns of the plot is not distinguishable in Fig. 11 (middle). Figure 7 (bottom) shows the h21, h22, and h23 pulse streams without the h20 stream. The network can be extended to convert any N-frequency stream with any size inter-pulse spacings.

An HTPE network that computes the differences between the corresponding inter-pulse spacings of two NM AP streams of equivalent form (i.e. the number of inter-pulse spacings and hence, the

number of pulses in a single period are equal in both streams). The Relative Difference Network (RDN) employs the SPC to strip off the individual inter-pulse spacings from FPWO encoded streams to prepare the them for the inter-pulse differencing operation. The use of the RDN for object recognition will be discussed in the projected progress section. The functionality of the RDN, however, will be described here due to its relation to the operation of the SPC and the data encoding technique of the FPWO.

An HTPE system for computing the relative difference between the parameters of any two NM pulse streams is shown in Fig. 8. The information contained in the pulse streams can represent any variable. In the case of the FORSE, the oscillatory pulse stream of the FPWO contains features that uniquely characterize specific objects. The NM oscillatory pulse streams are input to the SPCs that strip off the individual components (i.e. N inter-pulse spacings) of the stream by transforming them into the N phase shifts between N+1 equal frequency streams. The phase shifts are then converted to analog voltages by HTPE Phase-Lock-Loops (PLLs). The analog values corresponding to the respective components of the two streams are differenced. The resulting analog values are then reencoded in the inter-pulse spacings of an asynchronous, three-frequency, pulse stream.

The HTPE PLL, shown in Fig. 9, can be used for two primary functions: first, the PLL can be used to synchronize a data extraction or I/O system with a known frequency pulse stream, and second, the PLL can convert information encoded in a frequency or phase shift into an analog value. Synchronization is used to produce a reference for data transfer between HTPE and conventional digital systems and for the extraction of pulse-stream encoded information. The data conversion function of the PLL is useful for the extraction of frequency- and phase-coded information. The PLL can be used to directly measure the phase shift between two equal frequency signals or in conjunction with the SPC to measure single- or multi-frequency pulse streams. Recall, the SPC transforms an N-frequency stream into N+1 equal-frequency streams with relative phase shifts; since the inter-pulse spacings of the serial input stream are now phase shifts, they can be quantified by the PLL.

The PLL quantifies a phase shift by adaptively synchronizing two equal, single-frequency streams. Adaptation is performed by an HTPE adaptation module which increases or decreases the delay associated with one of the input streams until the two streams appear to be firing coincidentally inside the PLL. The analog value on the adaptation module after the system has settled is a measure of the phase shift between the two input pulse streams. The HTPE PLL operates as follows. Two equal-frequency streams "InA" and "InB" are input to the PLL through excitatory synapses "e1", "e2", and "e3" (the PLL determines the phase shift from InA to InB). InA fires e2 and e3, which are connected to hillocks "h1" and "h2". InB fires e1, which is connected to h1. The amplitude and duration of the e1 and e2 PSPs are set such that the two synapses must be fired within a certain time-window of each other to cause h1 to fire. This time-window width is determined by the delay on the e3 PSP input to h2. When a pulse in InA arrives at the PLL input, it causes e2 and e3 to fire. After the delay on e3 has elapsed, h2 fires and resets h1 and e2. If InB becomes active before h2, the PSP produced by e1 adds with the previously fired e2 PSP and causes h1 to fire. If InB becomes active after h2, the e2 PSP has already been reset, and therefore, will not fire h1. If h1 fires, it will also reset itself and e2, and therefore, h1 and e2 are reset each cycle since either h1 or h2 will fire each cycle (the asterisk signifies either h1 or h2 can reset h1 and e2). Adaptation module "a1" controls the delay on e3, and therefore, the time-window of h1. The value on the adaptation module is adjusted by the pulses produced by h1 and "h3". If h1 fires, it signifies that the phase shift between InA and InB is less than or equal to the h1 time-window, which causes the adaptation value to be changed such that the e3 delay is reduced by one time unit. The delay is continually reduced until h1 no longer fires. This lack of h1 activity is detected by the subsystem centered on h3. The subsystem is constructed to monitor the firing of h1 and h2 and to fire if h2 fires and h1 does not. The synapses "e4" and "i1" are fired simultaneously and have durations that are equal and much longer than any possible phase shifts. If h1 fires before h2, it

resets e4, and therefore, h3 will not fire that cycle. If h2 fires, it resets i1 allowing the e4 PSP to cause h3 to fire. If h3 fires, it resets e4 so that only one pulse is generated. The h3 pulse causes the adaptation module value to be changed such that the time-window is increased by one time unit. This modulation of the time window duration adaptively locks the PLL to the phase shift between the two input pulse streams. The analog value on the module after the PLL has settled is a measure of the phase shift.

Figure 10 shows the simulated change in the adaptation module value as the phase shift between the two input streams is varied. The adaptation module value starts at zero and rises until the two streams are locked (flat level between 0 and 5E3 time units), the unit value rises again as the phase shift is increased (positive slope between 5E3 and 10E3 time units), and then decreases as the phase shift is reduced (negative slope between 10E3 and 15E3 time units) until the PLL again settles. The PLL can track increasing and decreasing phase shifts for any frequency input streams since the system is driven by the input.

In the Relative Distance Network (RDN), shown in Fig. 8, each input stream is passed through a SPC and the phase shifts quantified by PLL arrays. The analog output voltages of the PLL arrays are input to a Differencing PLL (DPLL), which differences the values of the corresponding stream components (i.e. the differences between the first components of the streams, the second components, and so on). The DPLL is very similar to the PLL in function and structure. The differences are then reencoded in an NM stream by a FPWO.

Figure 11 illustrates the simulated behavior of the RDN. Two 3-parameter pulse streams are input to the system (represented by "h4" (top) and "h13" (second from top)). The inter-pulse spacings of h4 and h13 are (12, 22, and 17) and (22, 17, and 32) time units, respectively (for simplicity, all of the APs in the streams are set to unity). The relative differences of the components of h4 and h13 are encoded in the inter-pulse spacings of the output pulse stream (represented by "h173"), as shown in Fig. 11 (second from bottom and bottom). Figure 11 (second from bottom) shows the output waveform representing h13 - h4 (the differences are 10,-5, and 15 time units). The wide pulse width of the pulse leading the five time unit spacing signifies its negative sign. Figure 11 (bottom) shows the output waveform representing h4 - h13 (the differences are -10, 5, and -15 time units). The wide pulse widths are now preceding the -10 and -15 time unit spacings. The analog values representing the component differences need not be reencoded in a pulse stream, but can be used directly by local systems (the reencoding is useful if the information needs to be transported to another location for processing, but in this case it was done for illustration purposes). This network can compute the relative differences between any two inter-pulse encoded streams regardless of what information they encode, and can be expanded to more than two inputs. If two single-frequency pulse streams are input to the system, the resulting output is simply the difference between the two frequencies.

Simple and Complex Cell Networks

The specific behavior of biological and HTPE simple and complex cells is detailed in the first report. The primary attribute of these networks is their ability to perform complex spatiotemporal operations with uncomplicated processors [7]. The processors are very similar in form and operation with their primary differences being the activations of their connections and the source of their inputs. For example, a network that is sensitive to moving edges employs processors that are very similar to those used in a network that only detects stationary edges, but the activations on the connections between the motion-sensitive cells differ from those between the stationary cells. Additionally, the motion-cells receive input from the stationary cells, and therefore, are at a higher level in the hierarchical HVS. This homogenous structure of the networks make them easy to scale and implement in integrated circuitry. The dynamics of the HTPE simple and complex cell networks are examined below in terms of their connection dynamics.

The simple and complex cell networks, shown in Fig. 12, are two-dimensional (N x M) arrays of uniform HTPE processors with connections to their nine nearest-neighbor processors (each processor is included in its own set of nearest neighbors). Each cell receives an excitatory and an inhibitory input from both the input and output of itself and its neighboring cells and, therefore, has 36 input connections. Figure 13 illustrates the local connection scheme; excitatory and inhibitory synapses pairs connect the input (i) and output (o) of neighboring cells $C_{ij}$, where ($0 \leq i \leq 2$, $0 \leq j \leq 2$ - see Fig. 12). To understand how the local connections can produce a particular spatiotemporal behavior, the dynamics of the HTPE on which the networks are based must be understood. A detailed description of the HTPE motivation, structure, and functionality is provided as Appendix A and will be summarized here before proceeding with the network dynamics discussion. Additionally, IRSI has developed an HTPE simulator which facilitates rapid design of large-scale HTPE networks. The HTPE model used in the simulator is abstracted from the VLSI HTPE model in that it does not maintain the low-level VLSI circuitry details of the HTPE. It does, however, maintain enough circuitry detail to allow HTPE systems developed with it to map directly to HTPE circuitry which is fine-tuned using circuit simulation packages such as PSPICE and HSPICE. The simulator provides timely high-level design capability which is directly compatible with low-level design refinement. The simulator was used to produce the HTPE network simulations provided in this report (unless otherwise noted) and will also be discussed in Appendix A.

Figure 14a shows an illustration of Excitatory (top) and Inhibitory (center) PSPs (EPSPs and IPSPs, respectively) of the HTPE. These waveforms are characterized by their amplitude, delay, and duration, which are all independently controllable. The amplitude of the waveform is its height above or below the resting potential of the HTPE. The delay of the waveform is the time it takes for the waveform to appear after an AP has fired it. The duration is simply the time extent of the waveform after it has appeared. Figure 14a also shows the waveform (bottom) produced by summing the EPSP and IPSP. Figure 14b is a simulation of a single HTPE with one EPSP and one IPSP input. The voltage "v1.soma" is similar to the waveform of Fig. 14a (bottom). Notice that the HTPE begins to fire APs as soon as the EPSP arrives and the firing rate increases once the IPSP has elapsed. These time-dependent waveforms enable HTPE-based simple and complex cells to have a diverse range of behaviors.

Figure 15 shows the simulated behavior of a range of excitatory connections from a particular cell $C_{11}$ to each of its eight surrounding neighbor cells. As you move down the figure from top to bottom ($C_{00}$ to $C_{22}$), the delay and duration of the EPSP waveforms increase and decrease, respectively. Figure 16 shows a similar delay/duration gradient for IPSP connections between a cell and its neighbors. The use of these waveforms to produce spatiotemporal feature filters is illustrated in Fig. 17. Figure 17 shows four basic velocity filtering behaviors produced by HTPE complex cell networks (low-pass, high-pass, band-pass, and band-stop). Inputs to the complex cells are the outputs of the simple cells on the previous layer of the simple/complex cell hierarchy. Since simple cells are sensitive to stationary line and edge segments at particular orientations, their output signifies the existence of such a properly oriented segment. The local connections of the complex cells can be set such that the network is sensitive to line and edge segments moving at particular velocities and accelerations.

Figure 17 (top) shows an example of the band-pass filter behavior. For this operation, the amplitude of the EPSP connections from the input of $C_{11}$ to its neighboring cells is such that by itself the EPSP will have no effect on the neighboring cells. If the EPSP associated with one of the neighboring cells also becomes active during the duration of one of the EPSP connections, the two EPSP waveforms will add and the neighboring complex cell will become active. For example, in Fig. 17 (top) "e11-00" represents the EPSP connection from the input of $C_{11}$ to $C_{00}$, "in11" represents the input to $C_{11}$ (this signifies the simple cell driving $C_{11}$ has detected a line or edge segment at its orientation sensitivity), "in00" represents the input to $C_{00}$ (this input originates from

a simple cell that is sensitive to line and edge segments at an orientation similar to that of the simple cell driving $C_{11}$ but at a different location in the visual field), and "h00" represents the output of complex cell $C_{00}$. When in11 becomes active, it fires the e11-00 EPSP which by itself has no effect on $C_{00}$. If in00 becomes active during the duration of e11-00, the EPSP it produces adds with e11-00 and causes $C_{00}$ to fire (this is seen on the left side of Fig. 17 (top)). If in00 becomes active before or after the e11-00 duration, it will not cause $C_{00}$ to fire. Therefore, $C_{00}$ will become active only if the simple cell driving $C_{00}$ becomes active within a specified time window of the simple cell driving $C_{11}$ (i.e. a line or edge segment at a particular orientation is moving across the visual field within a particular range of velocities). Line and edge segments moving outside of the acceptable range are not passed to the complex cell layer outputs (i.e. they are not within the pass-band of the filter). The pass-band can be made narrow or broad depending on the resolution required and can be located at any position in the range of EPSPs producible with HTPEs (due to the design techniques employed in the HTPE, which are described in Appendix A, it can produce EPSPs and IPSPs with durations and delays ranging from nanoseconds to seconds which provides an extremely wide range of velocity sensitivities).

Figure 17 (second from top) illustrates the band-stop operation which is simply the inverse of the band-pass. The EPSP produced by in00 is sufficient to fire $C_{00}$ by itself. The connection from $C_{11}$, however, is inhibitory "i11-00" and will "block out" the EPSP produced by in00 if they occur simultaneously. Therefore, line and edge segments moving within a particular velocity range are not allowed to pass to the complex cell network outputs (Fig. 17 second from top, right). Segments with velocities outside of this range, however, are passed to the network output (Fig. 17 second from top, left).

A high-pass segment velocity filter is shown in Fig. 17 (second from bottom). The high-pass filter operation is similar to that of the band-pass with the delay of the EPSP reduced to zero (since the simulations are in the time domain, the pass-band of the high-pass velocity filter appears as a low-pass time filter i.e. high-velocities imply low transit times between spatial locations). Figure 17 (second from bottom, left) shows an acceptable segment velocity, while Fig. 17 (second from bottom, right) shows an unacceptable segment velocity. The low-pass segment velocity filter is similar to the band-pass and high-pass filters in operation. Figure 17 (bottom, left) shows an unacceptable high segment velocity while Fig. 17 (bottom, right) shows an acceptably low segment velocity.

The connections between a cell and its neighbors need not all be similar (i.e. the velocity selectivity of each of the connections can be different), and therefore, depending on the direction of the movement, the speed sensitivity will be different. If all of the connections are the same, the network is speed sensitive only because the direction of the movement is not important. Using different connections among the neighbors provides velocity sensitivity (speed and direction both are factors in the filtering operations). Regardless of the similarity or difference of the local connections, the same set of connections are applied across the entire complex cell network (i.e. the connection between a cell and a particular neighbor (i.e. $C_{00}$) is the same for all cells across the network). Therefore, the network will have the same speed and velocity sensitivities regardless of the location of the input stimulus to the network. A more complicated connection scheme allows for non-homogeneous local connections which provides acceleration sensitivity to the network.

Figure 18 illustrates the four possible velocity/acceleration relationships possible. In Fig. 18 (top) the local connections are such that the network is sensitive to line and edge segments moving with a specific initial velocity and a specific acceleration. The EPSP connection between $C_{01}$ and $C_{11}$ is band-pass with a minimal width pass-band (one time-unit). The EPSP connection between $C_{11}$ and $C_{21}$ is also band-pass with a minimal width pass-band, but the location of the pass-band has been shifted from the $C_{01}$-$C_{11}$ band (the acceptable segment velocity between $C_{11}$ and $C_{21}$ is

higher than the acceptable segment velocity between $C_{01}$ and $C_{11}$). Therefore, segments moving with a particular initial velocity and positive acceleration from $C_{01}$ are allowed to pass to the network output. If the second velocity sensitivity (i.e. that between $C_{11}$ and $C_{21}$) had been lower than the first, the acceleration sensitivity would have been negative (deceleration sensitivity). Other velocity/acceleration relationships are also illustrated in Fig. 18: variable initial velocity and exact acceleration (Fig. 18 second from top), exact initial velocity and variable acceleration (Fig. 18 second from bottom), and variable velocity and acceleration (Fig. 18 bottom). Once again if the velocity and acceleration connections are the same for all neighbors the network is speed and change of speed sensitive because direction plays no role in the filtering operation. One additional point should be made about providing both velocity and acceleration sensitivity in a single layer of complex cells: the selectivity is dependent on the initial position of the input stimulus. The acceleration gradient across the network is different depending on where the network first becomes active. For example, if the input stimulus appears at location A initially, the acceptable initial velocity and acceleration ranges from that point will be different than if the stimulus had initially appears at position B.

The way to remove this position-dependence is to employ a hierarchical network for velocity selectivity and subsequent acceleration sensitivity. The first layer will pass segments with the proper rate of change in position (speed or velocity) and the second layer, which receives its input directly from the first layer and the input, will pass segments with the proper rate of change in speed or velocity (speed change or acceleration). The networks layers will be identical in structure and will differ only in the values of local connections and the source of their inputs. Now regardless of the location of the initial input stimulus, the acceleration gradient will be the same. Multiple layers can be employed to define any complex velocity/acceleration gradient. Additionally, self-adapting connection schemes are being investigated to reduce the number of layers required for any arbitrary acceleration gradient to two.

In the HTPE network simulations thus far presented, the axon hillock component has been set to act as a thresholding pulse stream generator. Once the input to the HTPE exceeds the specified threshold, the HTPE begins to produce APs and continues to do so until the input drops below threshold. The pulse width and spacing ("refractory period") are both controlled by DC biases and are not affected by the amount by which the input exceeds the threshold. An alternative setting for the axon hillock causes it to act as a Voltage-Controlled-Oscillator (VCO). The pulse width and refractory period of the output stream are still controlled by DC biases but the refractory period is also a function of the amount by which the threshold is exceeded. The larger the input with respect to the threshold, the shorter the refractory period. The VCO behavior is useful for the conversion of an analog input signal into an analog-frequency pulse stream and can be used in the complex cell networks previously discussed.

A scheme that utilizes both the excitatory and inhibitory synapses on each connection is illustrated in Fig. 19 (we will refer to such a connection as a "compound connection"). Compound connections require the VCO behavior of the HTPE to allow the excitatory and inhibitory connections to each have an effect when used in conjunction. This requirement can be understood by examining the connections used in the four basic filtering operations. The low-pass, high-pass, and band-pass operations add two EPSPs to determine the pass-band of the filter (i.e. e11-00 + e11-11). The band-stop adds an EPSP and an IPSP to determine the stop-band of the filter (i.e. i11-00 + e11-11). To perform both operations there are three different input combinations that need to be distinguishable: 1) the case where the two EPSPs are coincident (i.e. in the pass-band of the band-pass, low-pass, or high-pass filter), 2) the case where the EPSP and IPSP are coincident (i.e. in the stop-band of the band-stop filter), and 3) the case where the EPSP and the IPSP are not coincident (i.e. the pass-band of the band-stop filter and the stop-band of the other filters). A compound connection is capable of performing one of three filtering operations that combine the

basic filters (band-pass, band-stop, low-pass, and high-pass). The three combinations are a band-pass/band-stop, low-pass/band-stop, and high-pass/band-stop.

Figure 19 shows two possible arrangements for a band-pass/band-stop compound connection. In Fig. 19 (top) the pass-band of the band-pass filter is signified by the dense packet of APs (this distinguishes the first case mentioned in the previous paragraph). Notice that before the packet of APs, the cell was also active but at a lower rate. This low rate region identifies the third case mentioned above, and the region of no activity identifies the second case. The EPSP and IPSP are also visible in the plot. Figure 19 (bottom) shows the same connection structure with a reversed order, the IPSP precedes the EPSP. An interesting behavior that is not illustrated here arises when the EPSP completely overlaps the IPSP with additional overlap on both ends of the IPSP, the two EPSP regions on either end of the IPSP form pass bands. In this case, the connection acts as a combination of a band-pass and a high-pass filter, a band-pass and a low-pass filter, or two band-pass filters. It is apparent that an incredible range of spatiotemporal filtering operations can be performed with HTPE complex cell networks.

If the VCO version of the axon hillock is to be employed in the simple and complex cell networks, the cells will require modification. In the previous simple and complex cell networks, the thresholding behavior of the axon hillock has been used and the occurrence of an AP signifies the occurrence of an event and is not a measure of the degree of the event. The VCO behavior, however, allows not only the occurrence of an event but also its degree (the existence of APs signifies occurrence and the firing rate encodes degree) to be represented. Processing in terms of occurrence is a series of temporal Boolean logic operations. These operations are simple to implement in HTPE networks but may require a large number of processors to provide adequate processing resolution. The large processor count raises a concern about the density limitations of conventional planar IC technologies. Processing in terms of degree allows a single HTPE to perform the functions that a set of temporal Boolean processors perform at the expensive of processor complexity. The number of processors is reduced but due to their increased complexity a density question may still arise. The pros and cons of both occurrence and degree processing will be fully investigated in the third work period.

Figure 20 illustrates the behavior two modified HTPE processors for processing in terms of occurrence and degree. Figure 20 (top and second from top) show the simulated behavior of a simple HTPE processor that operates on the firing rate and pattern of the input. The top figure show the input stream (the dark pulses are actually dense groups of five APs). The second figure shows the response of the processor to the input. The first input burst causes an increase in the processor input, "a1" in the second figure, (the pulse burst is effectively integrated to produce the net input to the processor) but is not sufficient to fire the cell. The net value decays over time at a specified rate, and therefore, the net input will return to a base level unless persistent input is applied. The second input AP packet further increases the net input because the effect of the previous packet has not completely dissipated. The increase is sufficient to cause output activity represented by "v1". A third and fourth packet follow closely and cause the output rate of the processor to dramatically increase. As the effects of the AP packets dissipate, the firing rate decreases until a late arriving packet causes a small surge in the output rate which dissipates quickly. The remaining plots in Fig. 20 show the simulated behaviors of a more complex rate processor that employs multiple inputs, synapses, adaptation modules (described in Appendix A). The simulations were included to illustrate the diverse encodings that can be produced with HTPE rate processors. To effectively use rate processing, specific coding techniques and networks need to be developed and evaluated. This issue will be addressed in the final work period and will be a significant objective of the early Phase II effort.

Since simple cells are sensitive to lines and edges at particular orientations they are analogous to conventional CNNs running edge extraction connections. A major difference between simple cells and CNNs is that a CNN extracts the edges in a image on a frame by frame basis. A frame is

applied to the CNN and as the network settles the edges appear on the output. The next frame is then applied and the process repeats. The output of the cycle at time t has no direct effect on the operation of the cycle at time t+1. Conventional CNNs and neural networks for that matter are memoryless systems unless previous cycle information is explicitly stored and reentered as input during future cycles or multi-layer structures with feedback are employed. On the other hand, HTPE-based simple cells are inherently time-based systems in which previous system activity directly affects the current and future behavior of the system. The input image is continuously processed; edge information passes through the simple cells to higher-level processes and changes in location or orientation are continuously detected and acted upon. This allows the temporal features of the input to be used for the enhancement of spatial features. For example, primitive spatial features such as lines that are moving at the same velocity may be attributes of a common higher-level feature such as a simple geometric surface or volume.

HTPE simple cell networks are identical to the complex cell networks in structure. The two networks differ only in the temporal settings of the local connections. The connections in the simple cell networks are temporally independent (delays are set to zero and durations are set to values that are longer than the basic processing rate of the system). Therefore, the temporal relationships between the inputs are not of interest in the simple cell networks and the processing is a function of spatial features only. The EPSPs and IPSPs appear to be different analog voltages. Since the inputs are triggered by the arrival of synchronous APs, the order of input arrival is still detectable in the cell output stream, and therefore, the HTPE simple cells are not entirely time-independent. This is not a disadvantage because the spatial processing of the cells is not restricted or hampered by the temporal behavior.

In conventional CNNs, different sets of connections are used to extract the different primitive features such as vertical and horizontal lines. The connection sets are usually suitable for one particular filtering operation only and do not extract any features other than the one they are tuned to extrac.. This has the disadvantage of requiring a number of connection sets to be applied to extract multiple features. Since a sequence of connection sets must be applied to each frame, the real-time application of these systems is not practically realizable. In biological simple cell networks, there are cells sensitive to lines and edges at all orientations across the entire field of view. Therefore, the different spatial filtering operations are all perform in parallel and the results of any operation are continuously available. This is a "brute force" approach and requires an enormous number of simple cells, and therefore, may not prove practical for implementation in currently available VLSI technology.

An alternative is to assign each neighbor connection a different amplitude such that any combination of input from the neighbors produce a unique net input. If the VCO HTPE is used, the output frequency of the cell will convey exactly what is going on spatiotemporally in the local neighborhood. Figure 21 (top) shows an amplitude assignment scheme for eight neighboring inputs. The amplitude of each of the EPSP inputs is such that any combination of them will result in a unique net input to the cell ("h-c.soma" in Fig. 21 (top)). The different frequencies produced in response to the net input uniquely identify the input combination, and therefore, the spatial behavior of the local neighborhood. Due to the time independent settings of the EPSP, no temporal information other than the arrival order of the inputs is present in the output pulse stream. In the remaining plots of Fig. 21, however, the EPSPs have not been made time independent (i.e. they can have variable delays and durations). This produces a wide range of complex spatiotemporal behaviors. The four bottom plots in Fig. 21 show the effects of varying the duration and input stream frequencies. For a fixed set of input frequencies the output pulse streams become denser as the EPSP duration is increased. This is logical since with longer durations, the probability of more inputs being active simultaneously are higher. Techniques for setting the simple cell parameters to elicit a particular encoding need to be developed, and the utility of the various encodings must be determined. Auto-tuning methods for determining the proper parameter settings will be examined during the next work period.

Disparity-Tuned Cells for Depth Perception

The next step in the visual processing of the FORSE is the determination of depth. Multiple mechanisms are used in the HVS for depth perception, some of which utilize the data provided by a single eye, while others utilize the information provided by both eyes. Mechanisms that use the information provided by a single eye include: occlusion, parallax, rotation of objects, relative size, shadow casting, and perspective [7]. Perhaps the most important mechanism for assessing depth is *stereopsis,* which is the fusing of the information provided by both eyes. Stereopsis is performed in the primary visual cortex by more sophisticated forms of simple and complex cells. These cells have the same properties as those mentioned earlier - line and edge sensitivity, movement sensitivity, directional sensitivity, and end-stopping - plus additional properties relevant to stereopsis. To keep things clear we will refer to the new forms of simple and complex cells as Disparity cells.

Disparity cells fall into four basic categories: disparity insensitive cells, tuned excitatory cells, near cells, and far cells. The latter three being collectively referred to as disparity tuned cells. To discuss the properties of the disparity cells, we must first introduce the concept of disparity. The following definition of *disparity* is taken directly from [7].

> "Suppose an observer fixes his gaze on a point P. This is equivalent to saying that he adjusts his eyes so that the images of P fall on the foveas, F (see Fig. 22a). Now suppose Q is another point in space, which appears to the observer to be the same distance away as P, and suppose QL and QR are the images of Q on the left and right retinas. Then we say that QL and QR are *corresponding points* on the two retinas. Obviously, the two foveas are corresponding points; equally obvious, from geometry, a point Q' judged by the observer to be closer to him than Q will produce two non-corresponding images Q'L and Q'R that are farther apart than they would be if they were corresponding (see Fig. 22b). If you like, they are outwardly displaced relative to each other, compared to the positions corresponding points would occupy. Similarly, a point farther from the observer will give images closer to each other (inwardly displaced) compared to corresponding points."

This lack of correspondence is referred to as disparity. This definition will now be used to describe the properties of the four categories of disparity cells.

*Disparity insensitive cells* are stimulated if a slit of light is swept across both retinas simultaneously. The position of this slit of light, however, has no significant effect on the output of the cell; only the fact that both retinas were stimulated is of consequence in disparity insensitive cells.

*Tuned excitatory cells* are stimulated maximally only when the stimuli to the two retinas falls exactly on corresponding parts of both retinas. The amount of horizontal disparity that can be tolerated before the cell's response disappears is a fraction of the width of the receptive field. Therefore, the cell fires if and only if the object is roughly as far as the distance on which the eyes are fixed. About half of disparity selective neurons are tuned excitatory [8]. They have maximal excitation for disparity of 6' and a range of 10' ±4'. They may be directionally selective or not and they may not respond at all to monocular stimulation over a narrow range of disparity. For foveal-tuned inhibitory neurons, maximal suppression occurs within ±6'. Tuned inhibitory neurons have their binocular responses suppressed within a narrow range of disparities. They give excitatory responses over a range of disparities of one sign and inhibitory responses over a similar range of opposite sign They are characterized by a steep response gradient from maximal excitation to maximal inhibition with the mid point of response activity at close to or at zero disparity.

*Near cells* fire maximally only when the object is closer than the distance on which the eyes are fixed (outwardly displaced points).

*Far cells* fire maximally only when the object is farther than the distance on which the eyes are fixed (inwardly displaced points).

The existence of tuned, near, and far neurons suggests that normal stereopsis is based on the activity of three populations of neurons preferentially activated by near zero, crossed, and uncrossed disparity. Neuronal networks sensitive to viewer centered coordinates and distances are able to recover the 3-D structure of the visual scene from binocular disparity and the fixation distance.

In the macaque monkey, nearly all the visual cortical neurons are binocular. Binocular neurons have two receptive fields, one for each eye, and their activity reflects the dynamic interaction of excitatory and inhibitory influences from each eye; thus, the responses of these neurons to binocularly viewed patterns may vary markedly depending on the receptive field characteristics and the spatiotemporal configuration of the stimulus in two eyes [8]. The receptive fields of binocular cortical neurons subserving central vision may be in exact spatial correspondence in the two eyes, or could have different relative positions, some field pairs having convergent disparities and other divergent disparities.

Cortical neurons (both simple and complex) measure binocular disparity by receptive field disparity. There is also speculation that the disparity cells may not represent absolute disparity but rather, disparity differences with respect to another point in the image. For each cell, a response profile can be constructed along the disparity domain over a range centered on the functionally optimal spatial superposition of the two monocular fields in space, and extends over a range of disparities that depends on the extent of field superposition and on the receptive filed size. These stereoscopic cells may then be thought of as matching primitives that correspond to isolated oriented edges [8].

Both simple and complex cells measure retinal image disparity by receptive field disparity and may operate in mechanisms of local stereopsis. Local stereopsis uses binocular disparity as the depth cue and localizes the depth of elements if the corresponding points of the elements are established without any ambiguity [9]. Global stereopsis involves determining (based on some global mechanism) the preferred set of depth localizations from among the possible localization sets. Binocular disparity detectors that are tuned to the same disparity values are pooled and have their simple receptive fields fall on different retinal positions. There might be a second type of global unit composed of simple disparity detectors tuned to different disparity values but inspecting the same retinal location. In [10], Juselz demonstrated that stereoscopic matches occur quite early in the processing of visual information and may precede form recognition. There are cells that signal the correct disparity avoiding the false matches in that process and is analogous to global stereopsis. Global constraints like continuity and uniqueness would be possible to implement in terms of inhibitory interactions among cells tuned to different disparities, and with overlapping receptive fields locations.

Simple positional disparity between the receptive fields in the two eyes may not be a sufficient mechanism for disparity detection of the correct matches [9]. At the cortical neurons, binocular matching is very precise, for these cells are capable of signaling the correct disparity of a few dots whose position over the cell's receptive fields changes 100 times/sec. Therefore a possible arrangement for these neurons may be based on sets of discrete and numerous receptive fields, or subfields, in the receptive field of one eye, "gated" with a similar set of appropriate positonal disparity on the other eye. False matches would be avoided by having cells with subfield disparity on the order of the associated monocular receptive field size but not larger. This scheme is computationally similar to the correlation of some function of the filtered image over the area

defined by the cortical receptive field. Another guess is that stereoscopic neurons may be matching a large set of monocular measurements of the image such as derivatives of the image intensities. A set of complex cells seems to be a logical solution for the correspondence problem. Disparity sensitive simple cells may represent and match monocularly strong isolated borders with a given orientation and their output may be used in the matching process.

The current HTPE disparity cells are very similar in structure to the previously examined simple and complex cell networks. The simple cell networks in which the cells employ the VCO HTPE and receive different amplitude connections from each of their neighbors, effectively encode the exact spatial orientation of their immediate neighborhoods. At any point in time, the output of the cell can be examined to determine the state of the cell and its eight immediate neighbors. This same graduated connection scheme is used to form Simple Disparity Cell (SDC) networks. The primary SDC inputs (e11-11 and i11-11) originate from a certain locations in the receptive field of either the right or left early visual processing systems (the early visual processing system is composed of the INSPIRES and simple and complex cell networks) and the neighboring inputs originate from a range of locations in the other early processing system. This range of locations includes points at particular disparities with respect to the primary SDC input. The output of the SDC at any point in time conveys which of the disparite inputs are active. The inputs to the SDC all originate from simple cells with the same line and edge orientation sensitivities, so the firing of the SDC indicates features that may correspond between the two early processing systems.

The same scheme is used in Complex Disparity Cell (CDC) networks to determine spatial correspondence using spatiotemporal correspondence. For example, the CDCs are configured in the same way as the SDCs with inputs originating in both early processing stages. The inputs to the CDC all originate from complex cells with the same line and edge orientation sensitivities, so the firing of the complex disparity cell indicates features with similar temporal features that may correspond between the two early processing systems. Any number of spatiotemporal characteristics can be used to determine correspondence between spatial features in the two early processing systems. The use of multiple spatial and temporal constraints to resolve correspondence between spatially disparite features will provide more reliable matching performance with lower mismatch rates. For example, lines appearing in both retinas which are moving at the same velocity are more likely to be matches than lines with different velocities.

Due to the programmability of HTPE networks, a particular HTPE-based disparity tuned cell can be made selective to a variable narrow disparity range or even swept through a wide range of disparity sensitivities. Additional HTPE circuitry can be used to develop disparity tuned cells that adaptively lock-on to the exact disparity between two objects, thus providing a more quantitative measure of object distance. The structure and simulated behavior of the HTPE disparity tuned cell networks will be provided in the final report. Additionally, the fusion of more than two spatial separate early visual processing systems will be examined in the final work period.

**Projected Progress:**

One of the primary issues to be resolved during the final work period is the integration of the FORSE with the other subsystems of the INAVISIONS. The INAVISIONS is a continuous-time, asynchronous processing system based on a fully-parallel, three-dimensional HTPE architecture. As is shown in Fig. 23, there are six subsystems that make-up the complete INAVISIONS including system #2 which is the focus of this effort.

1) **The INtegrated Saccadic Perception Image Resolution Enhancement System (INSPIRES).** The INSPIRES collects visual data asynchronously and performs preprocessing similar to the human retina and the early stages of the human primary visual cortex. The INSPIRES also provides several orders of magnitude enhancement in the spatial resolution of collected images.

2) **The Feature extraction and Object Recognition SystEm (FORSE).** The FORSE extracts and fuses the spatiotemporal features in the input visual data provided by the INSPIRES, and performs primitive and complex object recognition by employing networks motivated by the latter stages of the human primary visual cortex. The FORSE also performs binocular fusion to generate scene depth information.

3) **The SPatial Encoding And Representation System (SPEARS).** The SPEARS represents and encodes the dynamic characteristics of objects of interest in the visual field. Such characteristics include object position, velocity, acceleration, orientation, and current activity. This information is provided almost entirely by the FORSE. The SPEARS also computes the relative distance between encoded objects and maintains the link between the objects detected by the FORSE and those encoded by the SPEARS.

4) **The SeleCtive ATtEntion System (SCATES).** The SCATES selectively focuses the attention of the INAVISIONS based on significance tags assigned to objects in the task environment. The tag priorities determine the order in which the objects are attended.

5) **The EXpectation-guided perCEPTION System (EXCEPTIONS)** The EXCEPTIONS guides the overall behavior of the INAVISIONS by applying task-environment knowledge and past experience to best achieve the current task at hand. The EXCEPTIONS and SCATES work together to select and focus attention on objects in the task environment that are relevant to the current goals of the INAVISIONS.

6) **The NEUROMotor Actuator ConTrol System (NEUROMACTS).** The NEUROMACTS provides coordinated, dexterous control of robotic manipulators by fusing high-level expectations, real-time visual cues, and tactile feedback. The NEUROMACTS models the hand-eye coordination system of humans.

Each of these systems interacts with the FORSE, and therefore, the ultimate structure of the FORSE will be dependent on the requirements imposed by the other systems of the INAVISIONS. The integration of the FORSE into the complete INAVISIONS will be completed in the final work period.

Many of the issues relating to the finalization of the simple, complex, disparity, and higher-level feature extraction network designs are similar. First, biological systems employ incredible numbers of uncomplicated cells to perform complex tasks. Due to the limited space in conventional ICs, this approach may prove impossible. The other alternative is to use more complicated cells that can perform multiple tasks. The complicated cells, however, require more complex encoding techniques. The tradeoffs between complicated and primitive cells will be examined. Second, the optimal field distribution of the various cells with different sensitivities and ranges will need to be determined. Third, the organization and interfacing of the different cellular networks must be determined. Fourth, the requirements for the implementation of the complete system in three-dimensional ICs must be evaluated.

In the initial FORSE prototype, the high-level processing will be performed by a software algorithm to allow for timely development. As HTPE networks are developed to perform the high-level operations, the software system will pass the responsibility for those operations to the hardware system. For example, complex feature extraction and shape formation will initially be performed by the high-level software system, but complex cell networks similar to those used for edge and line segment enhancement are currently being developed and will ultimately be used to perform these operations. It is expected that a hierarchical organization of networks functionally similar to the simple and complex cell networks will perform tasks up to complex object recognition and interpretation. For example, the RDN that computes the differences between two

multi-frequency pulse streams can be used to perform time-based vector quantization for object recognition. If one of the inputs to the RDN is an exemplar and the other input is a test input vector, the output is the closeness of the two. Multiple exemplars can be compared against and the best match (defined by some programmable metric) can be selected with hierarchical Winner-Take-All (WTA) networks as the best match. Hierarchical WTA networks have been developed and will be detailed in the final report.

High-Level Depth Perception

Some depth determination mechanisms rely on expectations and knowledge of the surrounding environment. Expectation-guided perception and knowledge representation systems for the INAVISIONS are currently under development in other projects. These systems will provide high-level guidance to the object recognition system and other lower-level INAVISIONS systems. This guidance will assist the FORSE in the tasks of recognition and depth determination. The interface between the high-level systems and the object recognition system will be developed during the final work period as will the methods for implementing the various depth determination mechanisms (this will include the design and simulation of the required HTPE networks). High-level depth determination mechanisms include:

Occlusion - When an object is in front of another object and partially blocks its view we judge the front object to be closer and the back object to be farther away. [11]. An occluded region in a stereo pair is a set of pixels in one image representing points in space visible to one eye or camera only. These occur as parts of background immediately to the left or right sides of the nearby occluding objects, and are present in most natural scenes. Previous approaches to stereo ignored these unmatchable points or weeded them in a second pass [12,13]. Psychophysical evidence shows that the HVS, however, does exploit these clues [14,15] and in [16] a biologically-based neural network is discussed which constructs depth from occlusion.

Parallax - The relative motion of near and far objects that is produced when we move our heads from side to side or up and down is a powerful depth cue. The larger displacement of an object in the visual field implies that it is closer than an object with a smaller displacement.

Rotation of objects - The rotation of a solid object by even a slight amount can make its shape immediately apparent. The object shape or change in shape can therefore be used to judge depth. There are many kinds [17] of information such as edge, shading and motion in an image sequence when objects rotate.

Relative size - When the size of an object is roughly known, we can judge it distance by comparing its size in the image to its expected size. The expected object size is provided by the high-level expectation-guided perception system based on its knowledge of the object. This is a fairly strong clue for rigid bodies but may be ineffective in circumstances such as when the object is partially occluded. It is possible to implement HTPE comparator networks to compare expected and actual object sizes.

Shadow casting - A bump on the wall that juts out is brighter on top if the light source comes from above, and a pit in the surface lit from above is darker in its upper part. This depth information is useful for texture analysis. [18]. Several methods to identify shadows are found in the literature which provide greater understanding of the image. For example, in [18], Jiang has identified a three step method involving low-, mid-, and high-level process for identifying shadows. The low-level process extracts regions darker than its surroundings. The mid-level detects features such as the vertices on the outline of dark regions, self-shadows, and cast shadows in dark regions and the object regions adjacent to dark regions. The high-level process verifies a set of hypotheses and confirms the consistency of low-level detections.

Perspective - The image of parallel lines, like railroad tracks, as they go off into the distance draw closer together. This is an example of perspective, which provides powerful depth cues [7].

Each of these mechanisms can be used to provide depth information individually or in combination. Since each mechanism has its limitations, the combined use of all of them will provide more accurate depth determination. The use of these cues by the high-level depth perception system will be investigated in the final work period. The high-level depth perception cues will be used by the object recognition software system to assist in the recognition process.

Object Recognition

NNs are used in conjunction with various preprocessing networks for object recognition [19,20]. The problem with the present NN approaches which use either Kohonen Self-Organizing-Map (SOM), Multi-Layer Perceptron, or other NN classification paradigms is that they are usually the final stage of the classification. Their output cannot be processed by the next stage device due to the nature of their output. So the success of the NN approach boils down to the successful preprocessing circuitry and their application. Granted, conventional NNs have excellent parallel processing capability and discriminative ability but they need large numbers of training exemplars to be successful in an real world environment. Also, traditional NNs are computation intensive and require a large amount of training time. There are not many successful hardware implementations of NNs for real world applications. They are usually implemented as algorithms on serial machines, and thus, one of their inherent advantages - parallelism - is lost.

Another successful approach used is Vector Quantization [21] which employs either traditional quantization, handcoding, or NNs [22] for the quantization operation. NN coding is the best in the sense that they are not arbitrary features. The success of this approach is based on the selection of the feature vectors. The merging of these feature vectors for successful recognition of the object needs to be addressed.

Template matching [23] is an approach in which the stored image of an object is compared with that of the input image and the object is recognized. This approach becomes impractical in the case of a large number of objects, and the number of templates needed to be stored becomes too large to be practical. In fact, a number of references for such an approach can be found in Ronen [23].

Other approaches tried are hierarchical [24] or isolated feature matching [25]. The hierarchical feature matching starts from the low level to the high level where features are matched successively and the correct match is determined. This is like a complex network in which the important features to be matched are determined a priori and rule based approaches can be used.

In [26], Mumford argues that a combination of feature and template based approaches is needed for successful object recognition. They cite an example of Carpenter and Grossberg approach. A new object activates a set of feature level modules bottom-up. They activate, in turn, a set of higher-level categories of objects, and in a top down fashion, templates of prototypes of objects are produced. The low level tries to match these to the scene and triggers all mismatches. The higher level stores data on the range of allowable variations for all objects ,and on receiving the triggers modifies the definitions, and therefore, creates adaptive templates.

Our approach to object recognition emphasizes the process of interpretation using other system components to extract three-dimensional feature information from the image. The object recognition component of the system has a number of advantages given it by the processing capabilities of the other components. Briefly, they are:

1. Parallel, Data Driven - All processing is done in parallel, from the image collection through feature extraction, up to the recognition component. The recognition component is performed in software for the initial prototype systems. However, it remains logically a data-parallel processor.

2. Time-Based Encodings and Processing - Because the encodings of features are time-based, it is natural that the feature recognition performed by the simple and complex cells (and later more complex feature recognition networks) use time differences to disambiguate difficult line and feature segmentation images. The system prioritizes input by assigning higher frequencies to input that has higher intensities, thereby making the encodings for such input as short as possible. This, in turn, allows the information to be processed more quickly than lower priority input that has longer encodings.

3. Consistent Spatial Map - The SPEARS subsystem of INAVISIONS maintains a consistent spatial map of the objects in the environment, including those that are in the field of view and those that the system knows about but are currently not visible. This allows the feature and object recognition subsystems to be constantly aware of the (possibly changing) relationships between primitive and complex features in the field of view.

4. Selective Attention - The task of object recognition is simplified by the SCATES component of the INAVISIONS that focuses attention on the objects in the field of view of greatest interest. The features of these objects will be priority encoded and will therefore be recognized and operated on much faster than the remaining elements. This provides the feature and object recognition systems with an automatic filtering mechanism, where only a small portion of the entire visual field will require processing for recognition.

5. Feature Recognition - The local dynamics of the simple and complex cell networks define what feature information can be extracted from the image. These networks provide the object recognition with time-varying features that describe the objects in the visual field that are currently being attended to.

Figure 24 (top) shows a simplified diagram of the current configuration of the object recognition subsystem of the INAVISIONS. Feature recognition from the (SCATES modulated) input is accomplished by the simple and complex cell networks, with the aid of disparity information. Feature information such as point, line, and arc segments are rectified in spatial coordinates by the SPEARS subsystem. More sophisticated local dynamics of complex cell networks can recognize spatially sensitive features such as corners and line junctions. As these networks are developed, those functions that are now duplicated in the software-based object recognition system will be removed, and the logical place of the software system will be after the new network component (see Fig. 24 bottom).

Because the object recognition component is provided with a description of the input image in terms of the features extracted and their spatial relationships, the assignment of semantic labels to different aspects of the image necessarily takes on a different character than the object recognition systems mentioned previously. Instead of the commonly used segment and search paradigm, where the entire image is segmented and a model database is searched for matching components, our method can be viewed as more of a bottom-up, grammar-based approach.

Recall that in general, a grammar will consist of a set of terminal and non-terminal symbols. Objects in the grammar are defined by a set of rules that specify how non-terminal and terminal symbols can be combined to form an image of the object. Thus, a non-terminal feature may be a corner, which can be recognized by the intersection of two lines. The objective of this approach is to allow objects to be described with a large number of features. This allows many relatively inexpensive operations to be performed that aid in the disambiguation of several interpretations,

instead of expensive operations that can only distinguish the positive occurrence of one object (template matching).

Figure 25 shows the hierarchical nature of the grammar-based approach. Primitive features that are recognized by the simple and complex cell networks correspond to the terminal symbols. Primitive features are extracted by simple and complex cell networks, which are then used to determine more complex features. These can be combined arbitrarily many times in order to develop recognizers for more and more sophisticated features. When the final level is reached, at least one interpretation of the input as an object is made.

When multiple objects are considered to be valid interpretations of the input image, the grammar can be used to determine characteristics that can distinguish between the objects under consideration. The simple and complex cell networks that constitute the characteristics can be accessed in order to generate disambiguation information if the primitive features have already been computed, or control information can be passed to the cell networks to configure them to compute the required primitive features.

The software grammer-based object recognition high-level depth perception techniques and algorithms will be refined during the last work period and tied-in with the developed low-level hardware systems. A complete description and overview of the FORSE will also be provided in the final report. Additionally, the objectives of the Phase II effort will be developed and presented in the final report.

## References

[1] M. DeYong, R. Findley, and C. Fields, "The design, fabrication, and test of a new hybrid analog-digital neural processing element," IEEE Transactions on Neural Networks, Vol. 3, No. 3, 1992.

[2] M. DeYong, T. Eskridge, and C. Fields, "Temporal signal processing with high-speed hybrid analog-digital neural networks," Jour. of Analog Integrated Circuits and Signal Processing, Vol. 2. Boston: Kluwer Academic Publishers, 1992.

[3] M. DeYong, T. Eskridge, and A. Palmer, "A coupled-grid neural network retina for real-time visual processing," Proc. IEEE 35th. Midwest Symposium on Circuits and Systems, Washington, D.C., 1992.

[4] C. Fields, M. DeYong, and R. Findley, "Computational capabilities of biologically realistic analog processing elements," In: J. Delgado-Frias (Ed.) *VLSI for Artificial Intelligence and Neural Networks*. New York: Plenum, 1992.

[5] J. Maunsell and W. Newsome, "Visual processing in monkey extrastriate cortex," Annual Review of Neuroscience, Vol. 10, 1987.

[6] E. DeYoe and D. Van Essen, "Concurrent processing streams in monkey visual cortex," Trends in Neuroscience, Vol. 11, No. 5, 1988.

[7] D.H Hubel, *Eye, Brain, and Vision*. New York: W.H. Freeman and Company, 1988.

[8] G.F. Poggio and T. Poggio, "The analysis of stereopsis," Annual Review of Neuroscience, Vol. 7, pp. 379-412, 1984.

[9] B. Julesz and M. Hill, "Global stereopsis: cooperative phenomena in stereoptic depth perception," In: R. Held, H.W. Leibowitz, and H.L. Teuber (Eds.) *Handbook of Sensory Physiology* VIII - Perception, pp. 215-256. New York: Springer-Verlag, 1978.

[10] B. Julesz, "Binocular depth perception of computer-generated patterns," Bell System Technical Journal, Vol. 39, pp. 1125-1162, 1960.

[11] P.N. Belhumeur and D. Mumford, "A Bayesian treatment of the stereo correspondence problem using half-occluded regions," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 506-511. Los Alamitos, CA: IEEE Computer Society Press, 1992.

[12] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," Science, Vol. 194, pp. 283-287, 1976.
[13] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 139-154, March, 1985.
[14] K. Nakayama and S. Shimojo, "Da Vinci stereopsis: depth and subjective occluding contours from unpaired image points," Vision Research , Vol. 30, No. 11, pp. 1811-1825, 1990.
[15] B. Anderson, Personal Communication, October, 1991.
[16]P. Sajda and L.H. Finkel, "Object segmentation and binding within a biologically-based neural network model of depth-from-occlusion," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 688-691. Los Alamitos, CA: IEEE Computer Society Press, 1992.
[17] J.Y. Zheng and F. Kishino, "Verifying and combining different visual cues into a 3D model," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 777-780. Los Alamitos, CA: IEEE Computer Society Press, 1992.
[18] C. Jiang and M.O. Ward, "Shadow identification," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 606-612. Los Alamitos, CA: IEEE Computer Society Press, 1992.
[19] E. Mjolness, "Visual grammars and their neural nets," In: J.E. Moody et. al. (Eds.) *Advances in Neural Information Processing Systems* 4, pp. 461-67. San Mateo, CA: Morgan Kaufmann Publishers, 1992.
[20] J.I. Minnix, "Rotation and scale invariant pattern recognition using a multistaged network," Proc. SPIE Visual Communications and Image Processing, Vol. 1606, pp. 241-251, 1991.
[21] K. Flaton, "Multi-stage object recognition using dynamical link graph matching," Proc. SPIE Applications of Artificial Neural Networks II, Vol. 1469, pp. 137-148, 1991.
[22] K.S. Min and H.L. Min, "Fast identification of images using neural networks," Proc. SPIE Applications of Artificial Neural Networks II, Vol. 1469, pp. 129-133, 1991.
[23] R. Basri, and S. Ullman, "Linear operator for object recognition," In: J.E. Moody et. al. (Eds.) *Advances in Neural Information Processing Systems* 4, pp. 452-459. San Mateo, CA: Morgan Kaufmann Publishers, 1992.
[24] P. Seitz and G.K. Lang, "Using local orientation and hierarchical spatial feature matching for the robust detection of objects," Proc. SPIE Visual Communications and Image Processing, Vol. 1606, pp. 252-259, 1991.
[25] L. Ke and Y. Jing-yu, "Recognizing industrial parts using 2-D shape features and geometry parameters," Proc. SPIE Applications of Digital Image Processing XIII, Vol. 1349, pp. 388-392, 1990.
[26] D. Mumford, "Mathematical theories of Shape: do they model perception," Proc. SPIE Geometric Methods in Computer Vision, Vol. 1570, pp. 2-8, 1991.

Appendix A References

[27] R.F. Lyon and C. Mead, "An analog electronic cochlea," IEEE Transactions on ASSP, Vol. 36, No. 7, 1988.
[28] J. Lazzaro and C. Mead, "A silicon model of auditory localization," Neural Computation, Vol. 1, No. 1, pp. 47-57, 1989.
[29] C. Mead and M. Mahowald, "A silicon model of early visual processing," Neural Networks, Vol. 1, No. 1, pp. 91-97, 1988.
[30] M. DeYong, R. Findley, and C. Fields, "Computing with fast modulation: experiments with biologically-realistic model neurons," Proc. 5th. Rocky Mountain Conference on AI, Las Cruces, NM, pp. 111-116, 1990.
[31] R. Findley, M. DeYong, and C. Fields, "High speed analog computation via VLSI implementable neural networks," Proc. 3rd. Microelectronic Education Conference and Exposition, San Jose, CA, pp. 113-123, 1990.

## Appendix A

HTPE Motivations - Development of the HTPE was initiated in 1989 with support from the NASA Innovative Research Program (Grant NAGW 1592). The goals of the project were to develop realistic models of the electrophysiological behavior of typical spiking neurons at a level of description that allowed investigation of different time-dependent models of channel conductances, and to use these models to investigate processing of time-dependent signals. Scaling to the volt-nanosecond operating domain and modeling both passive membrane conductances and channel populations with discrete MOS devices allowed flexible models with minimal device counts to be developed and simulated using SPICE. This modeling methodology can be contrasted with that of Mead and colleagues [27-29], who have employed primarily operational transconductance amplifiers to model neurons, neuron components, and small circuits in the mV - ms operating domain. Circuits developed with the HTPE are generally smaller, lower-power, less noise-sensitive, and much faster than circuits developed with operational amplifiers.

A principal motivation behind the development of the HTPE was to realistically model a biological neuron without sacrificing speed and computational power. Biological realism in neural modeling is motivated both by the goal of understanding the behavior of biological nervous systems, and by the realization that biological neurons are complex, versatile signal processing devices that are evidently well-suited to a very large variety of computational tasks. Neurons are hybrid analog/discrete devices, in which inputs are processed by the time-dependent convolution of relatively slowly-varying PSPs, and outputs are transmitted over long distances by fast, relatively loss-free action potentials APs. The integration of continuous, asynchronous analog input processing with discrete, pulse-encoded communication allows neurons to make use of time and phase differences between signals arriving in real time to represent both temporal and spatial information. It also allows neurons to exchange information in times much smaller than their internal processing times, hence breaking the communication bottleneck that hobbles many massively parallel systems. The combination of high-speed, discrete communication and versatile analog computation makes neurons ideal for many time-dependent signal processing applications. The HTPE, which we have developed over the past three years, is the first artificial neuron to explicitly model the generation and processing of both PSPs and APs in hardware [30,31].

The HTPE represents a qualitative advance over conventional digital and analog processing elements. In conventional digital systems, all processing is synchronous (clocked). Synchronization imposes a fixed lower limit on the temporal resolution of processing, and forces a discrete representation at no better than the processing resolution on continuous input signals. Conventional analog processors accept continuous input, but embody the assumption that transient responses to inputs can be neglected. In the most common hardware PE, incoming signals are summed and then convolved with a sigmoidal transfer function, as is done in standard software simulations of ANNs. By only using steady-state responses in processing, these systems impose an effective upper limit equal to the relaxation time of transients on the overall temporal resolution of the system. Biological neurons are sensitively dependent on transient responses of ion channels to fluctuations in membrane potential - the AP itself can be viewed as a transient response - and both accept and generate signals asynchronously. The HTPE is similarly a fully asynchronous PE that allows information to be encoded in both the transient and steady-state components of time-varying signals [30,31].

HTPE Technical Approach - While many of the properties of biological neurons are important in developing robust ANNs, some are artifacts of the neuron's implementation in organic matter. The HTPE has been designed to model the computationally advantageous properties of neurons while making full use of the speed and compactness of silicon implementation. Signal timing in the HTPE has been linearly scaled to take advantage of this speed while maintaining the relative shapes and timing of both APs and PSPs. VLSI input voltages are typically on the order of a volt; therefore the signal voltages have been scaled from the millivolt range of neural signals to the volt

range of MOS devices. The operating range of a typical neuron is roughly -80mV, just below the Nernst potential for $K^+$ ions, to roughly +60mV, just above the Nernst potential of $Na^+$ ions. This range has been scaled linearly to the [0,5] volt range of MOS devices. This maps the resting potential of approximately -60mV to 1 volt and the typical threshold voltage of -40mV to 1.7 volts. The operating range for temporal relations in neurons, which is in the ms range, has also been scaled to the ns range of VLSI devices (approximately 1ms per 10ns in the current implementation).

Neurons are generally highly specialized, both electrochemically and geometrically, to particular signal processing functions. The HTPE, in contrast, is a "generic" model neuron that can be customized either by adjusting device sizes at the fabrication stage or by adjusting DC control voltages during operation. The design of the HTPE reflects a computational interpretation of neural architecture that views modulatory connections between neurons as the primary mechanism of real-time programming in neural circuits, and supports a flexible set of modulatory and learning mechanisms.

Anticipated Benefits of the HTPE - The primary advantage of the HTPE is its capability to process temporal information. Conventional ANNs are based on the assumptions that each input vector is independent of other inputs, and the job of the neural network is to extract patterns contained within the input vector that are sufficient to characterize it. For problems of this type, which amount to local spatial pattern recognition, a network that assumes time independence will provide acceptable performance. However, in a large class of signal analysis problems the input vectors are not independent and the network must process each vector with respect to both its own temporal characteristics and its relations to previous vectors. Network architectures that assume time independence are typically unwieldy when applied to a temporal problem, and require additional inputs, neuron states, and/or feedback structures. Although temporal characteristics can be converted into activation levels, this is difficult to do without losing information that is critical to solving the problem efficiently. Networks that assume time dependence have the advantage of being able to handle both time dependent and time independent data. The HTPE is ideal for developing networks of this type. Particular advantages of the HTPE for time-dependent signal processing include the following:

- The modeling of the continuous/asynchronous dynamics of biological neurons allows the HTPE to detect ultrafine temporal characteristics of the input signals, such as frequency and phase differences, smaller than the minimum switching time of the MOSFET devices on which it is based.

- The HTPE is designed to exploit the full nonlinear range of the MOSFET devices on which it is based. This has two consequences: first, the HTPE has an extremely wide temporal behavior range resulting from the use of all four modes of MOSFET operation (i.e.., cutoff, sub-threshold, saturation, and ohmic); second, the additional circuitry used in conventional systems for biasing and non-linearity compensation is not required, thus reducing the device count and VLSI area of the HTPE. Custom VLSI layout further reduces the HTPE layout area by ensuring that each MOSFET device performs a necessary function and that it is the minimal size required to provide adequate drive capabilities.

- The behavior of the HTPE is fully controllable through DC biases, which makes systems of HTPEs easily configurable. Neural systems typically require locally-dense globally-sparse interconnection schemes, which make them ideal for VLSI implementation.

- Parallel systems are inherently noise immune. This in conjunction with built-in adaptation mechanisms allows the HTPE and its systems to be very robust by absorbing system noise and imperfections.

- Several HTPE characteristics make it a very low power device: first, the system is asynchronous, which reduces the peak power requirements of the HTPE with respect to conventional digital systems in which all devices switch at the same time; second, since the quiescent state of the HTPE is rest it has much less static power dissipation than conventional analog systems, which constantly dissipate power to maintain a particular quiescent state; and third, minimal device count and size reduces the overall power requirements of the HTPE.

- Time-based, asynchronous data encoding provides several distinct advantages: first, encoding of analog values on the time axis provides immunity to signal noise, which plagues conventional analog systems; second, due to the fine temporal sensitivity of the HTPE, large amounts of information (as compared to conventional digital encodings) can be encoded in a short time delay between to APs; and third, if priority or frequently used information is encoded as short time delays it will automatically be processed at a higher rate, due to the asynchronous nature of HTPE processing.

HTPE Functional Description - Three types of chemical synapse circuits respond to APs at their inputs by providing current to or drawing current from a summing node that models the soma (cell body). These currents are time-varying waveforms and model the alpha-function form of PSPs. Excitatory PSPs (EPSPs) provide current to the soma, Inhibitory PSPs (IPSPs) draw current from the soma, and Shunting PSPs (SPSPs) maintain the soma at its resting potential, and can either supply or draw current depending on the current state of the soma. EPSPs, IPSPs, and SPSPs are produced by excitatory, inhibitory, and shunting synapse circuitry, respectively. The amplitude, duration, and delay of the PSPs are independently controlled by DC bias voltages applied to the synapse circuits. The soma node is modeled as a Resistive/Capacitive (RC) network with a membrane resting potential source. The soma capacitance represents the parasitic capacitances of all of the devices connected to the soma node. The soma resistance is modeled by a single NMOS transistor. The time constant of the RC network sets the upper bound on the analog computation rate of a given HTPE. The actual computation rate of an HTPE is determined by the DC biases on its components. The soma activity is monitored by the axon hillock, which generates AP streams that convey the current state of the HTPE. These AP streams feed the synapses of neighboring HTPEs, as well as synapses connected to the HTPE's own soma node. The hillock is a thresholding device with a hard-threshold below which the output is inactive. Above the hard-threshold is a time-dependent threshold region, where the delay before AP generation is proportional to the voltage above the hard-threshold. In this region the HTPE output frequency is a function of the soma voltage level, i.e. the hillock functions as a non-linear Voltage-Controlled Oscillator (VCO). As the input voltage approaches the instantaneous-threshold level (roughly one volt above the hard-threshold), the delay between output APs is reduced to its lower limit, which is the minimum propagation delay through the hillock circuitry. There are also three independently adjustable parameters of axon hillock behavior: threshold/delay level, AP pulse width, and the minimum separation between APs (the "refractory period"). In both the synapses and axon hillock, the parameter variations with respect to the DC biases are very non-linear, and offer an extremely wide dynamic range of behavior. All timing parameters of the HTPE are adjustable from approximately 1 nanosecond to at least 1 second. The amplitude parameters of the HTPE are adjustable through the entire zero to five volt range. Figure 26 shows the effect of simultaneously firing a shunting synapse and an excitatory synapse on one node and a shunting synapse and an inhibitory synapse on the other. Notice, the EPSP begins to spike shortly after the AP arrives but is quickly pulled down by the SPSP.

Adaptation modules, which either increase, decrease, or shunt a particular DC bias voltage a delta amount in response to an input AP. The modulating APs can originate from any point in the network, including the HTPE to which the adaptation module is attached. The delta bias change produced by an AP entering an adaptation module has different effects depending on where the bias is currently located on the bias/parameter non-linearity (the non-linear transconductance of a

single MOS transistor). The module adjusts the effect of each incoming AP by modeling the bias/parameter non-linearity and using negative feedback to either increase or decrease the effect of the individual incoming APs. Adaptation modules provide a mechanism for programming the behavior of the HTPE and its systems. Self-modulation provides the basis for a learning mechanism that loosely models hippocampal long-term potentiation (LTP) via the NMDA receptor mechanism.

HTPE Simulator Description - The large-scale HTPE system simulator was developed to reduce the front-end design time of larger HTPE networks. Before the simulator was developed, all HTPE system simulations were performed at the MOSFET circuit level. This was extremely time consuming for large systems. The structure of HTPE networks and the HTPE design itself served to further complicate these low-level simulations. HTPE networks employ large amounts of feedback which can confuse the iterative numerical solution techniques employed by circuit simulators. The individual HTPEs also have internal feedback. It was stated earlier that the HTPE employs unconventional design techniques that allow the individual MOSFET devices to operate in all four modes of operation (cutoff, subthreshold, saturation, and ohmic). This puts an additional burden on the circuit simulator, forcing it to check the mode of operation of each device at each time step and modify the equations of the devices that changed modes. Since the operation modes of the devices are constantly changing, the simulation speed is greatly reduced.

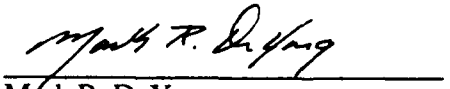
A C-based simulator was developed to overcome these limitations. The simulator qualitatively models the APs and PSPs waveforms of the HTPE without the low-level details of the associated circuitry. The simulator maintains enough detail, however, to allow the developed systems to map directly to a circuit implementation. The simulator allows us to simulate the "logic" of an HTPE network before the circuitry issues are addressed. This allows for the rapid development of large HTPE networks. For example, designing and HTPE network with the simulator will tell us the general configuration of the system and what components are needed to perform the desired processing. The low-level details can then be determined with a circuit simulator. We intend to continually refine the simulator so as to further reduce the low-level design time (by adding more circuitry detail), but not at the expense of any significant increase in the high-level design time.

The time axis in the simulations is divided in terms of generic units of time, simply referred to as *time units*. For example, APs in a pulse stream may have a pulse width of three time units and the stream may have a period of 15 time units. With the current MOSFET implementations of the HTPE a time unit can be roughly viewed as less than or equal to one nanosecond.

**DFAR 252.227-7036: CERTIFICATION OF TECHNICAL DATA CONFORMITY (MAY 1987)**

The contractor, Intelligent Reasoning Systems, Inc, hereby certifies that, to the best of its knowledge and belief, the technical data delivered herewith under Contract No. N00014-93-C-0118 is complete, accurate, and complies with all requirements of the contract.

Date: 4 August 1993



Mark R. DeYong
President and Principal Investigator

**Fig. 1:** Pulse Stream Encoding Methods.

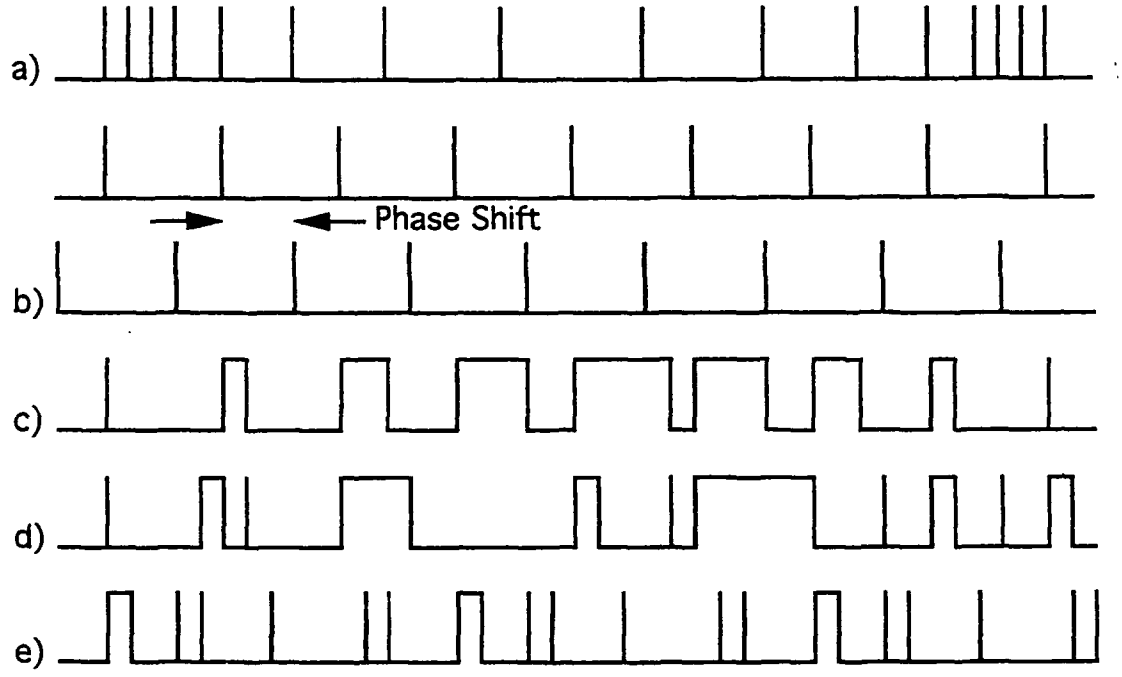


**Fig. 2:** Non-Linear vs Linear Encoding Techniques.

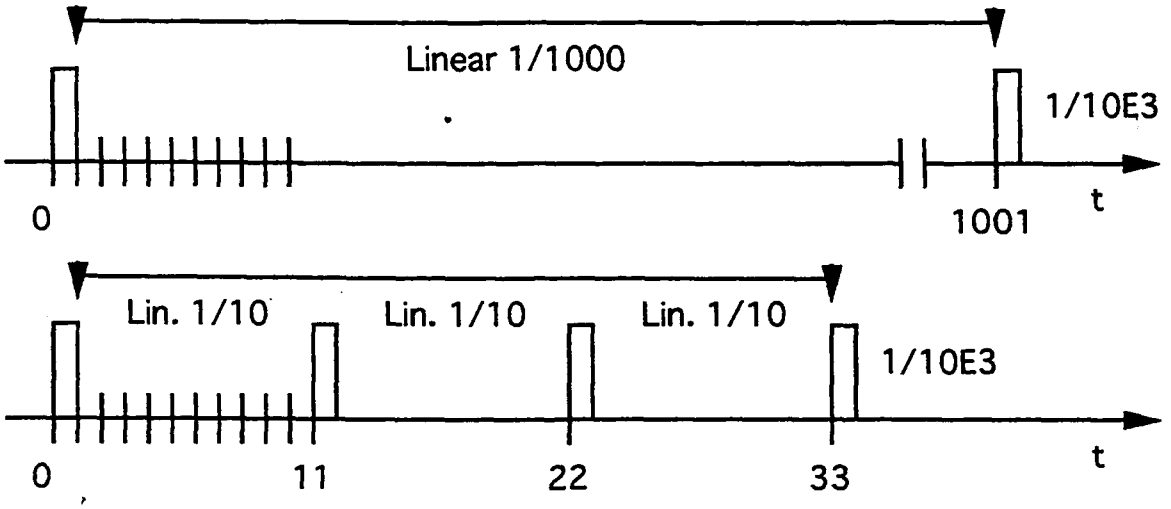


**Fig. 3:** HTPE Bias/Parameter Non-Linearity.

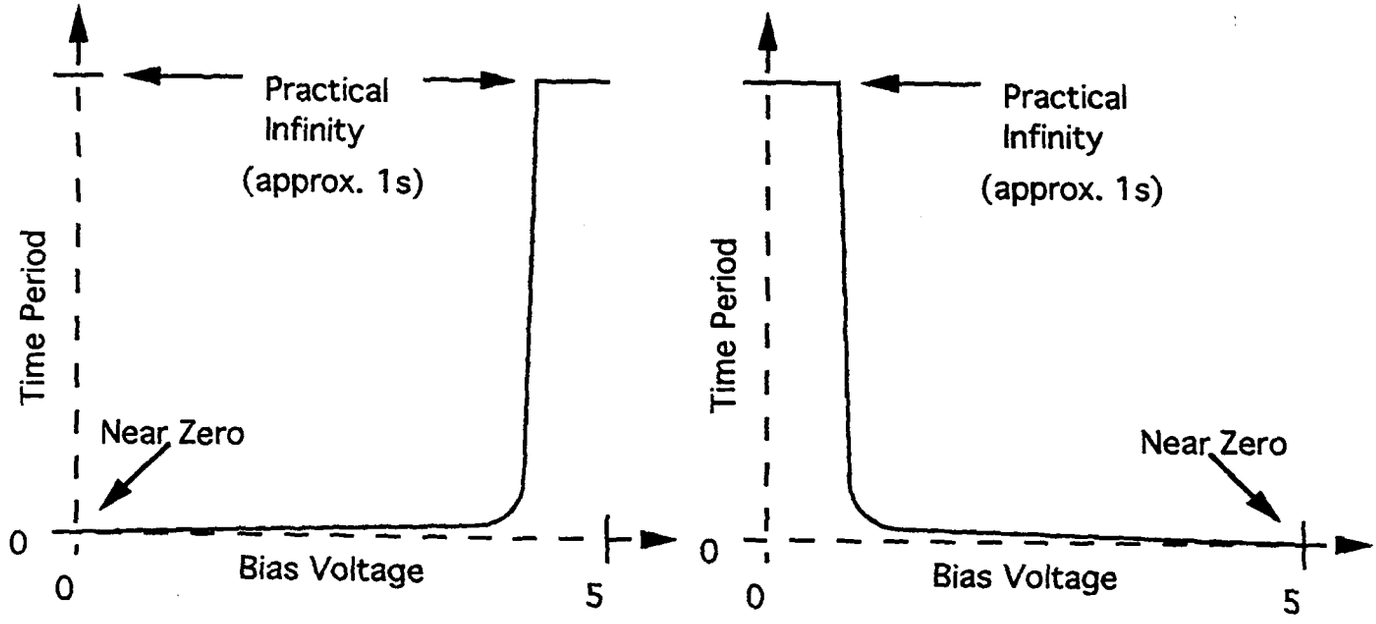


**Fig. 4:** HTPE Frequency/Pulse Width Oscillator.

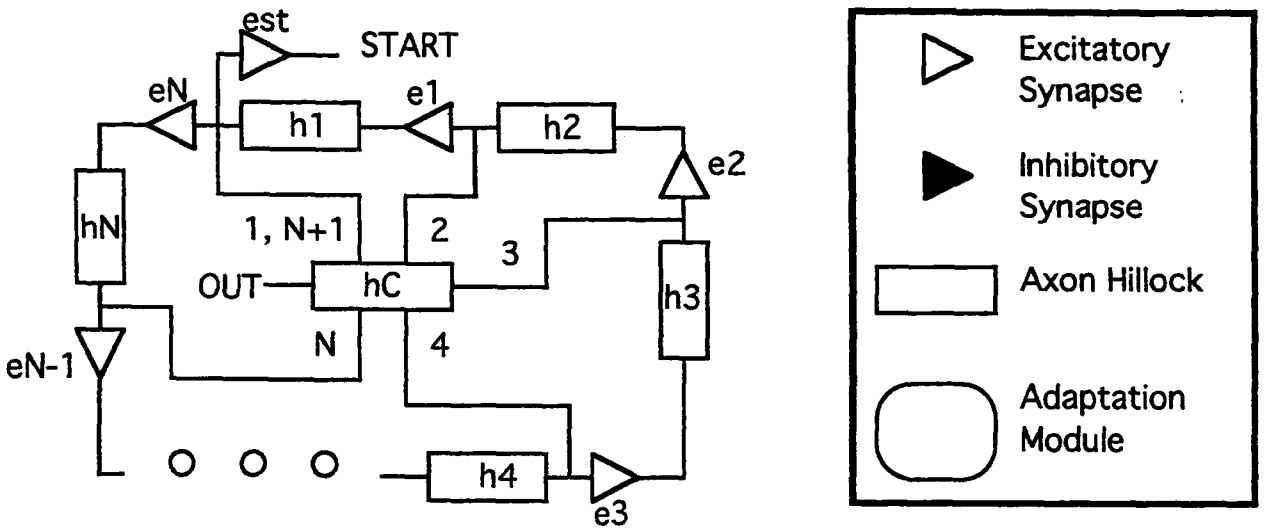


**Fig. 5:** Simulated Behavior of the HTPE Frequency/Pulse Width Oscillator.

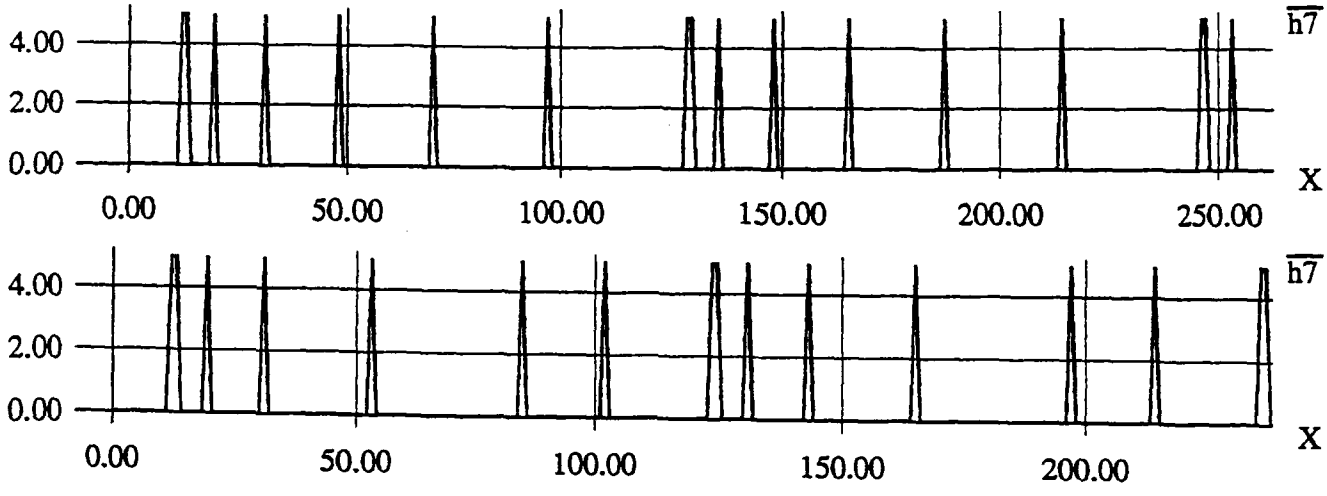


Fig. 6: HTPE Serial-to-Parallel Converter.

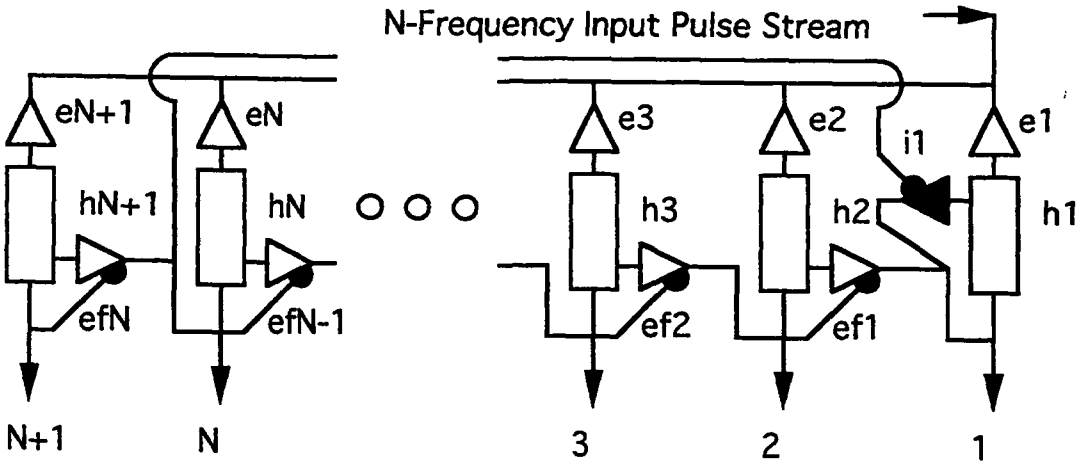


**Fig. 7:** Simulated Behavior of the HTPE Serial-to-Parallel Converter.

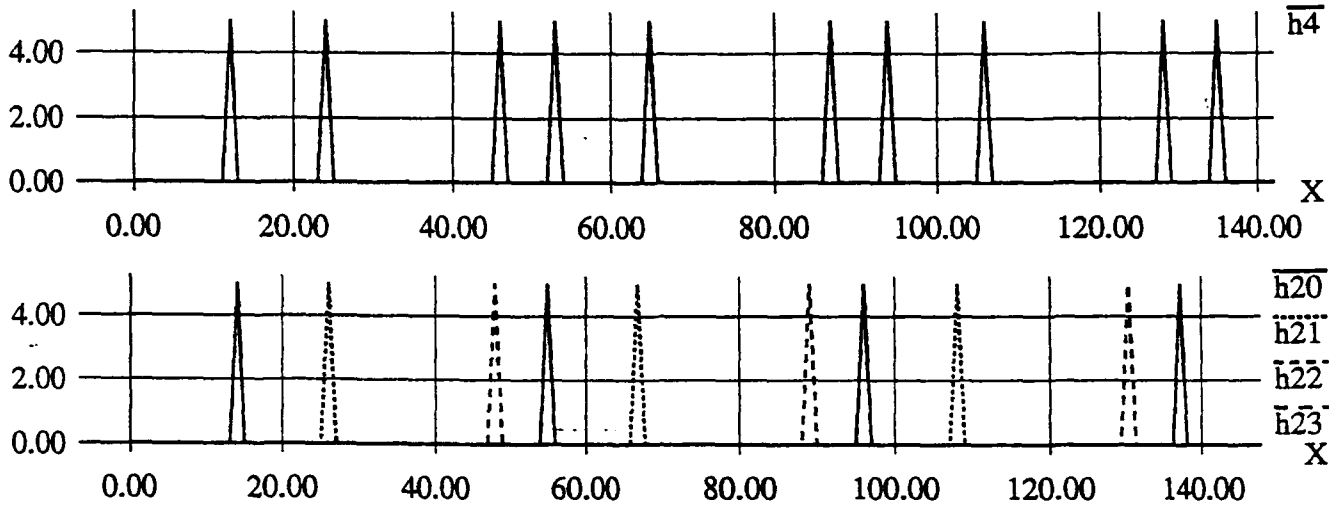


**Fig. 8:** Block Diagram of HTPE Relative Difference Network.

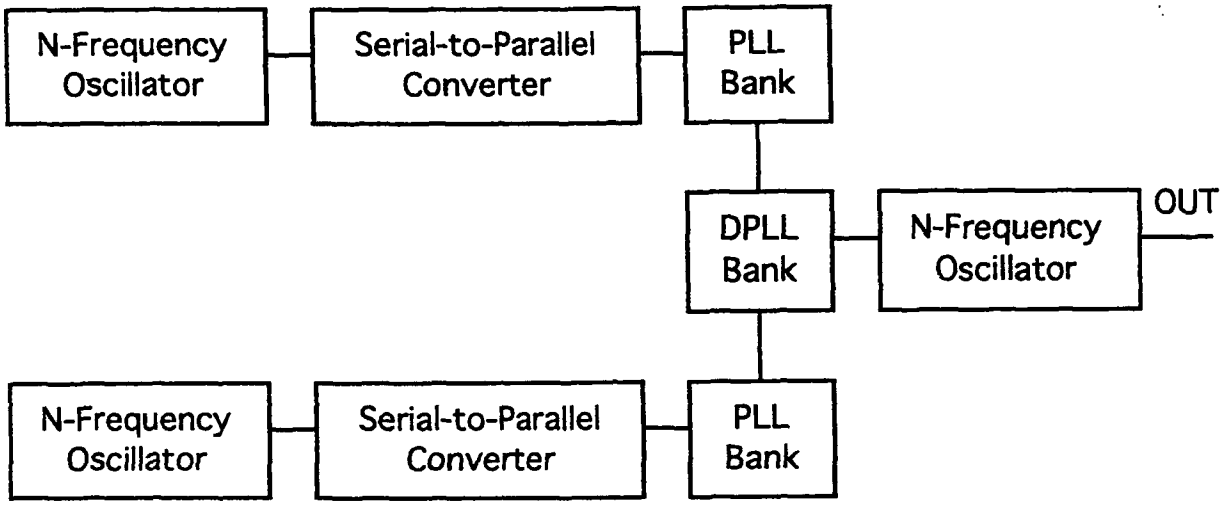


**Fig. 9:** HTPE Phase-Lock-Loop.

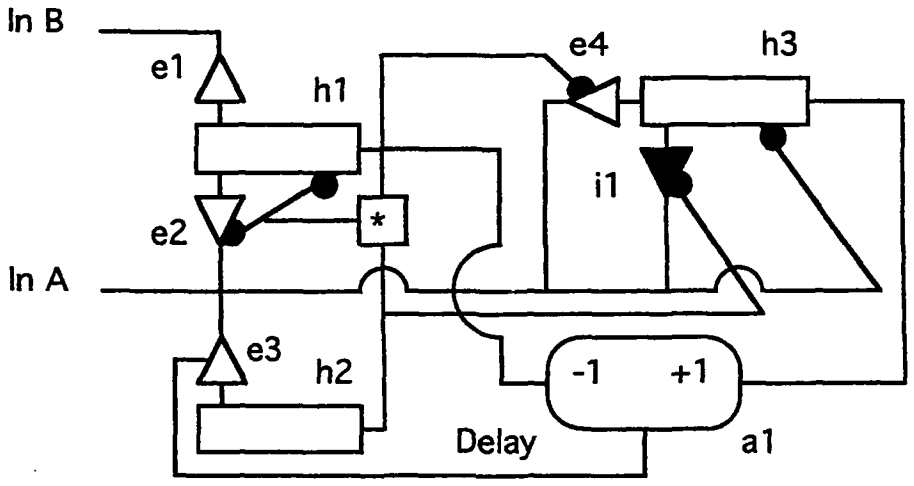


**Fig. 10:** Simulated Behavior of the HTPE Phase-Lock-Loop.

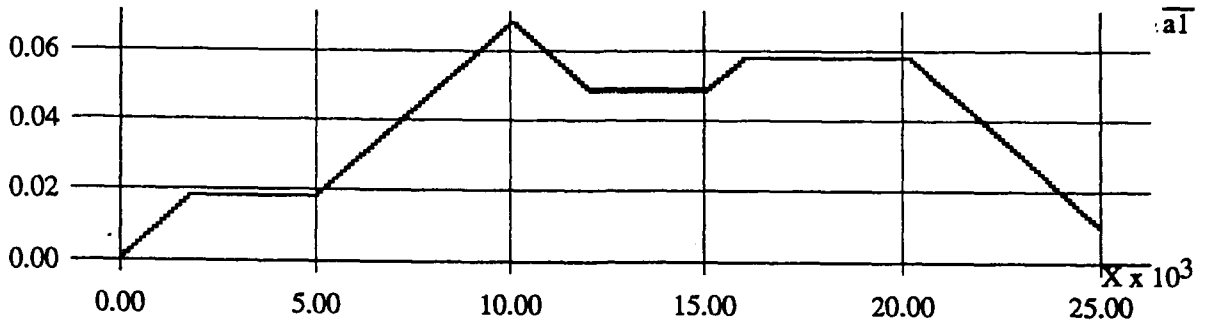


**Fig. 11:** Simulated Behavior of the HTPE Relative Difference Network.

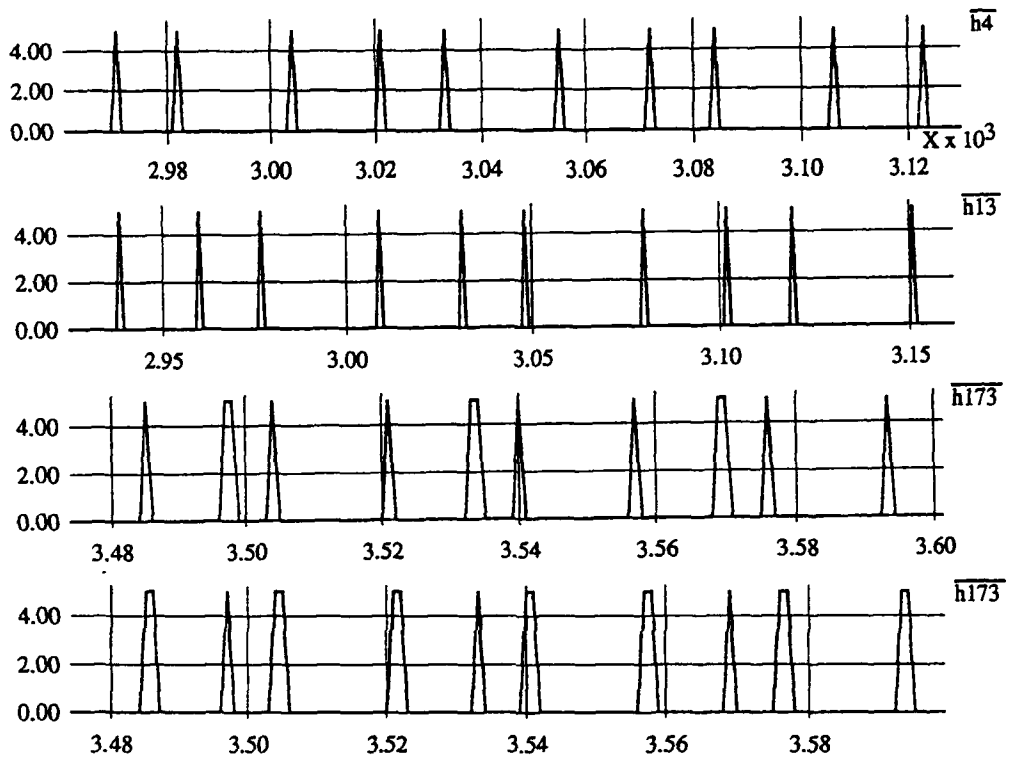


**Fig. 12:** Structure of the Two-Dimensional Simple and Complex Cell Networks.

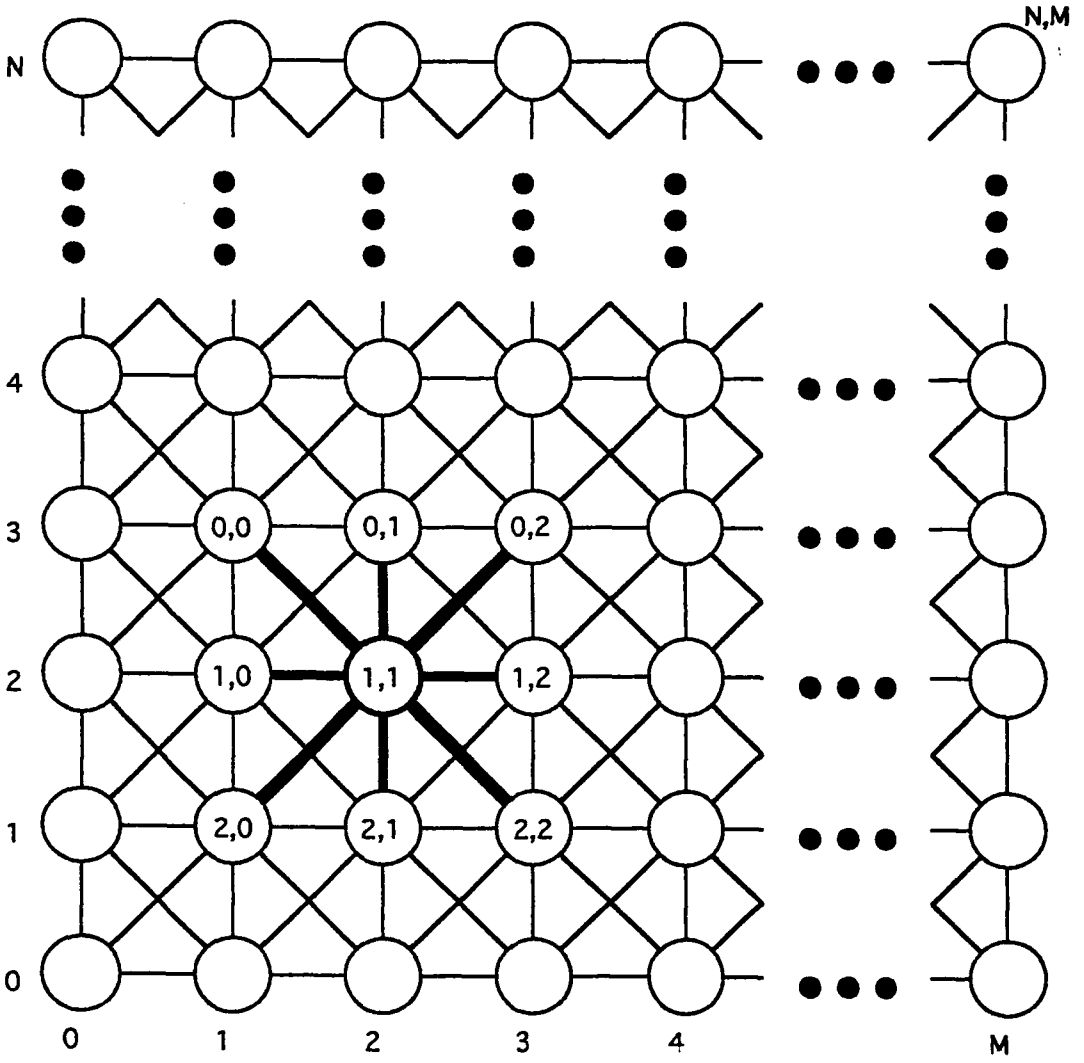


**Fig. 13:** Nearest-Neighbor Connection Scheme.

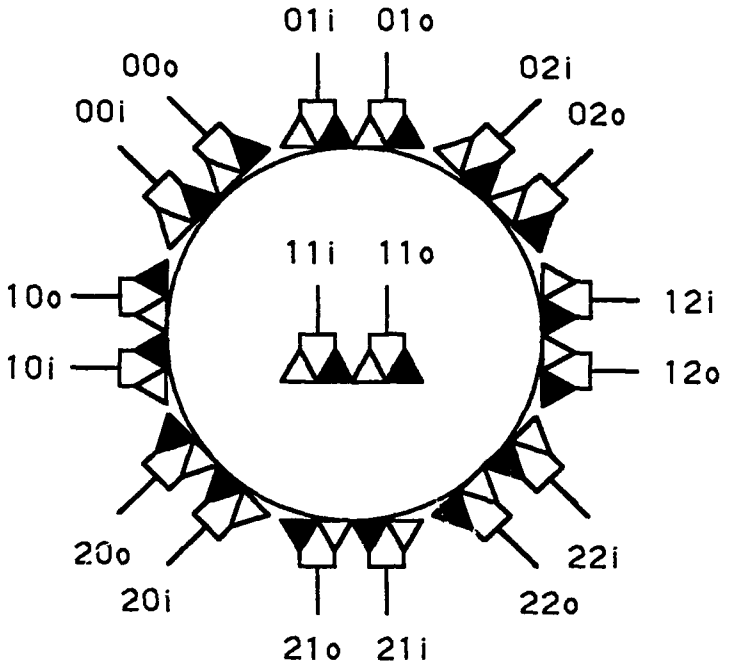
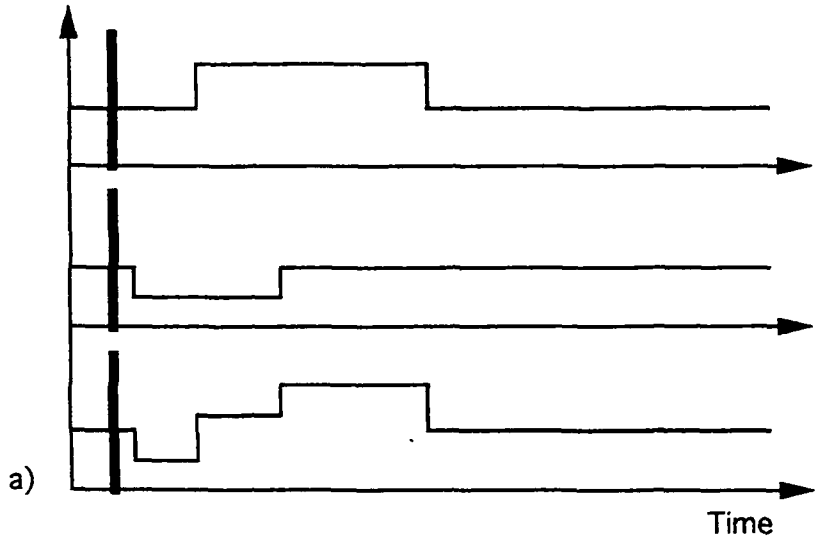
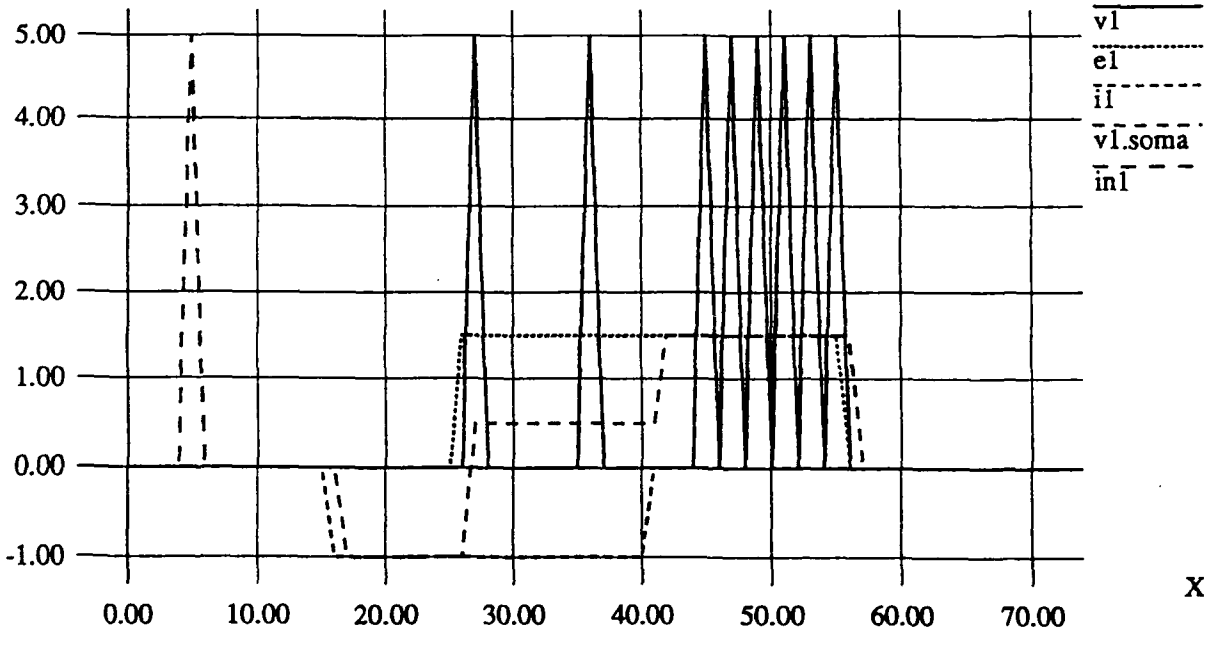


**Fig. 14:** Basic HTPE Waveforms and Operation.




b)

Fig. 15: EPSP Nearest-Neighbor Connections.

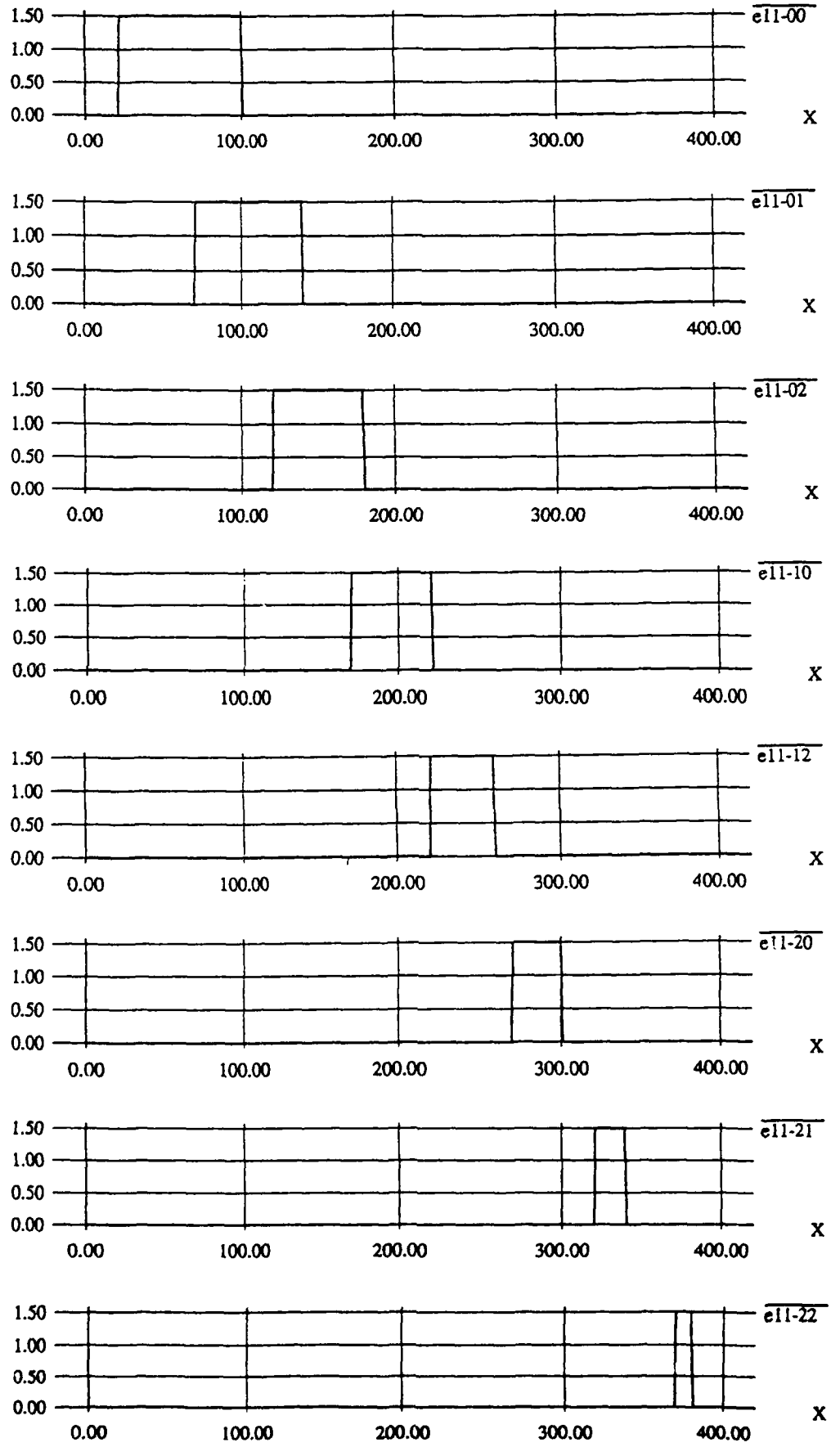


**Fig. 16:** IPSP Nearest-Neighbor Connections.

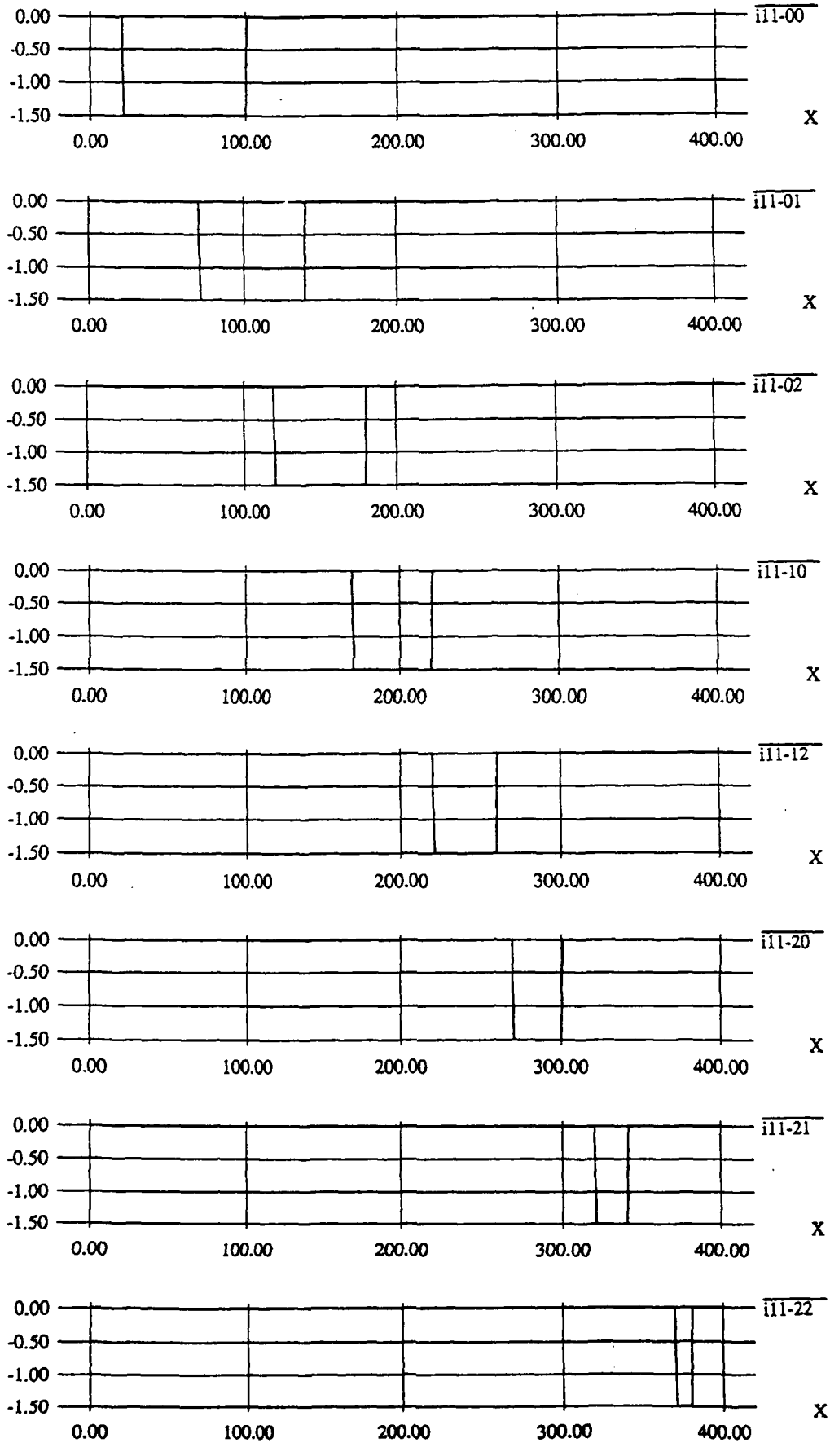


**Fig. 17:** Basic Velocity Filtering Operations of Complex Cell Networks.

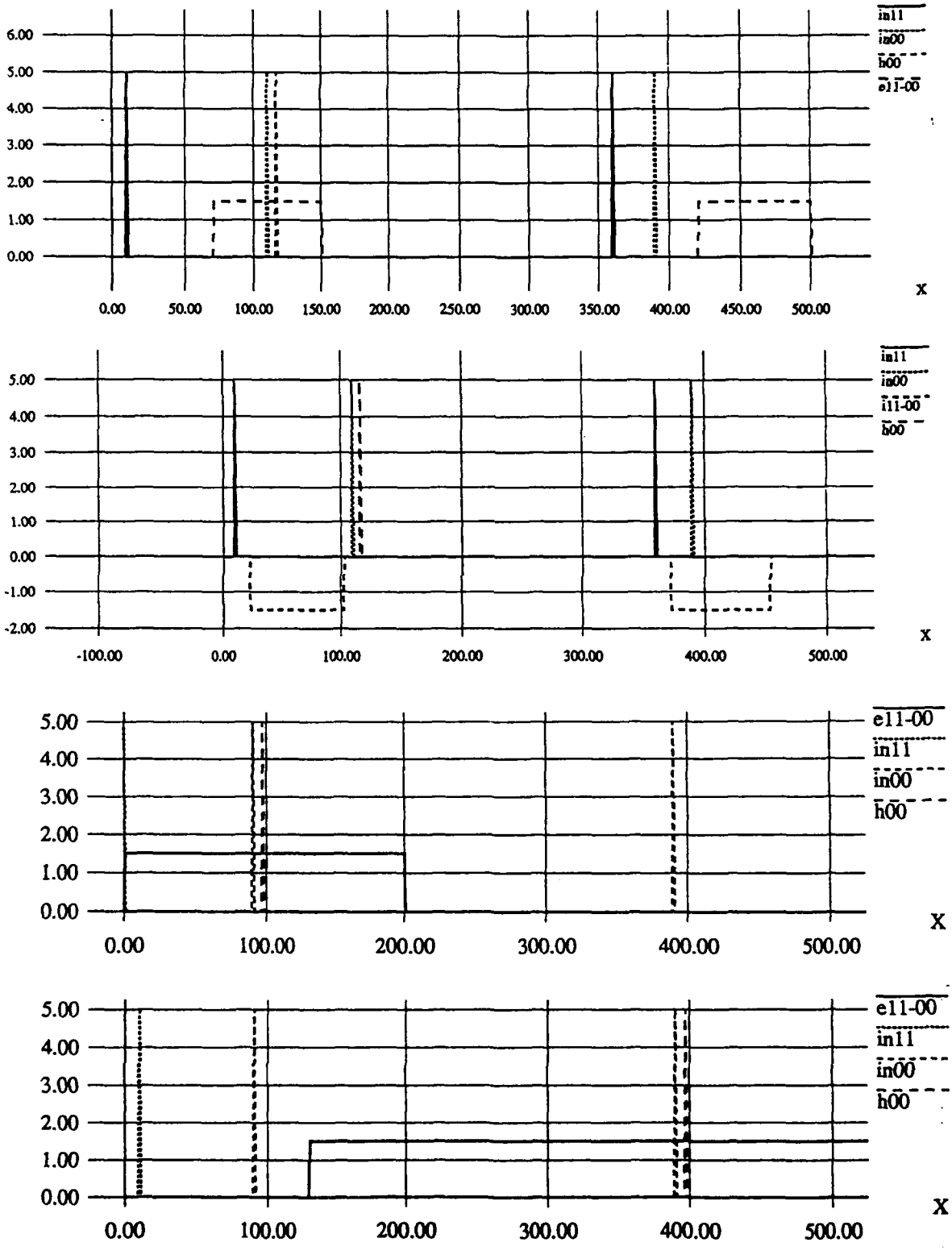


**Fig. 18:** Velocity and Acceleration Filtering Operations of Complex Cell Networks.

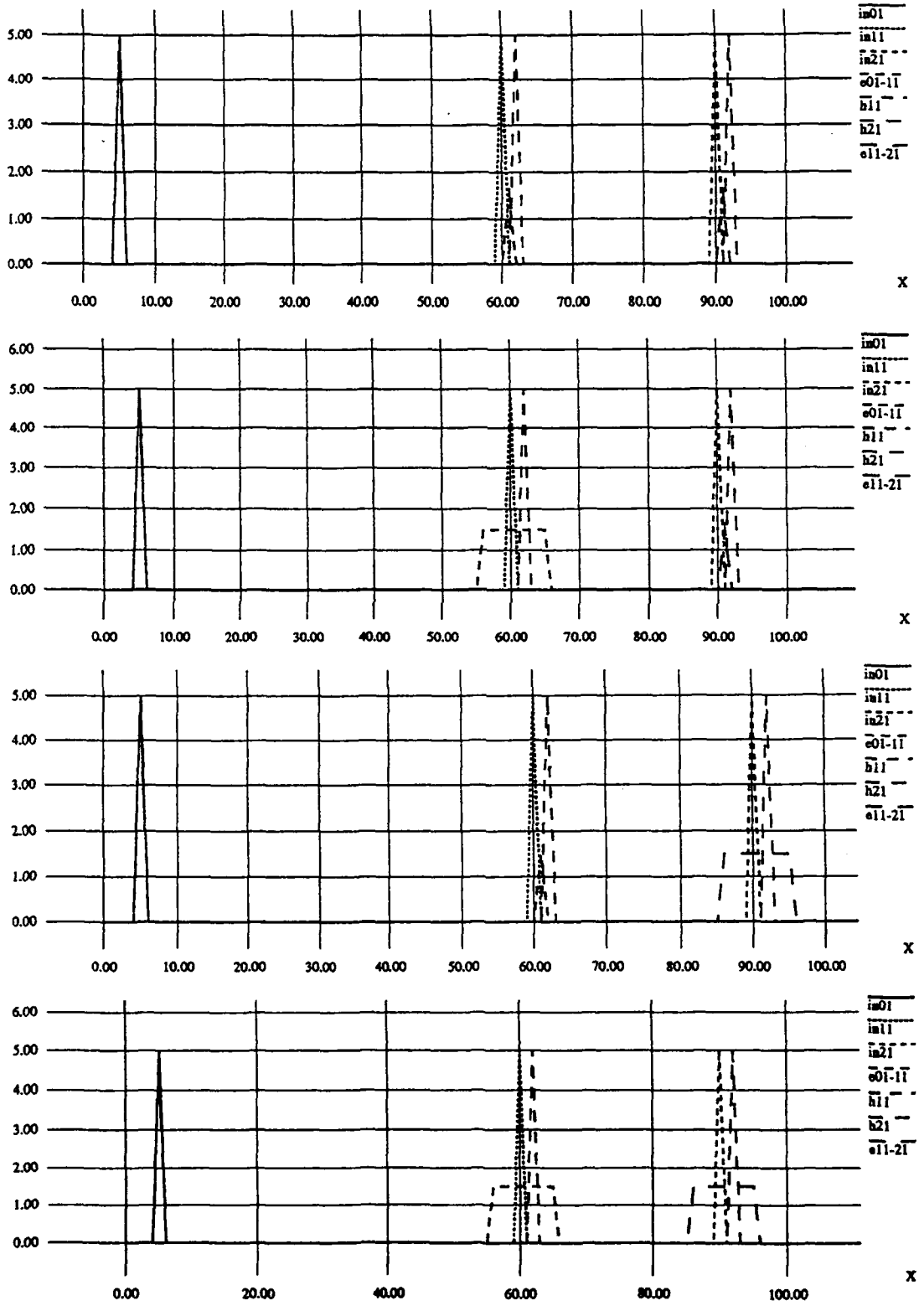


**Fig. 19:** Compound Connections of Complex Cell Networks.

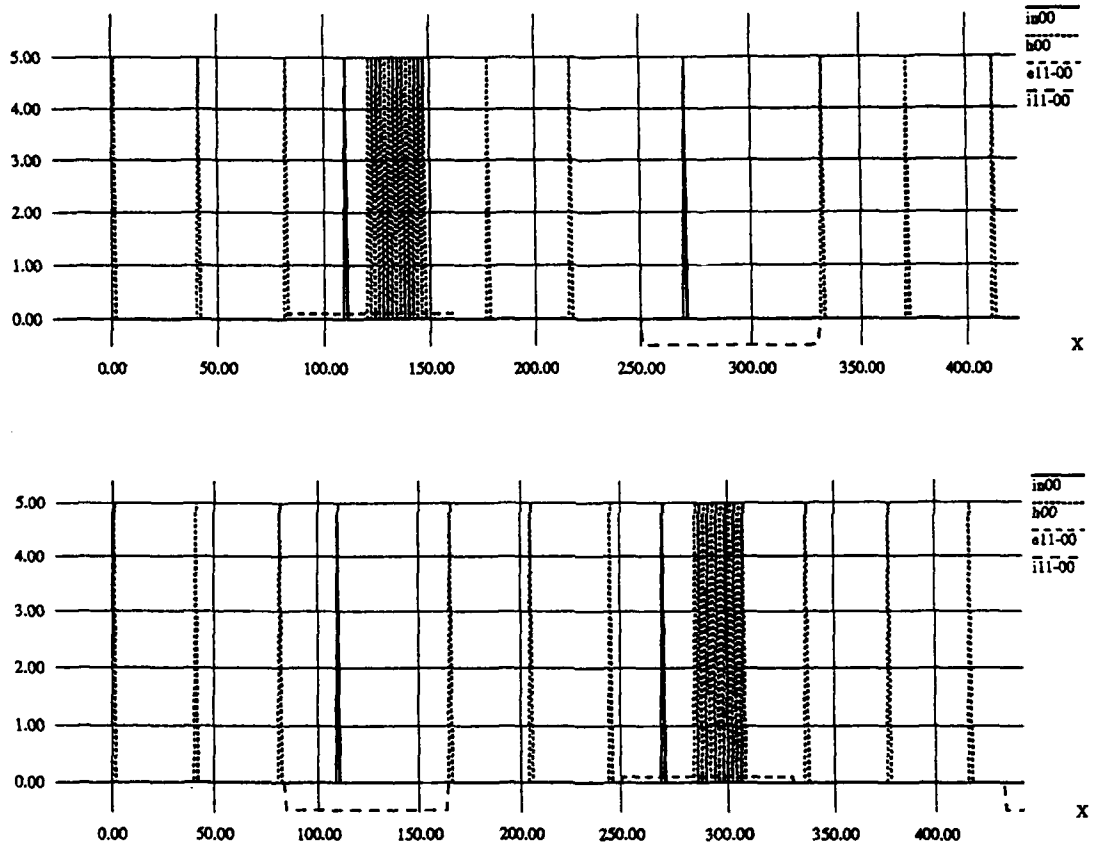


**Fig. 20:** Degree Encodings Produced by Modified HTPE.

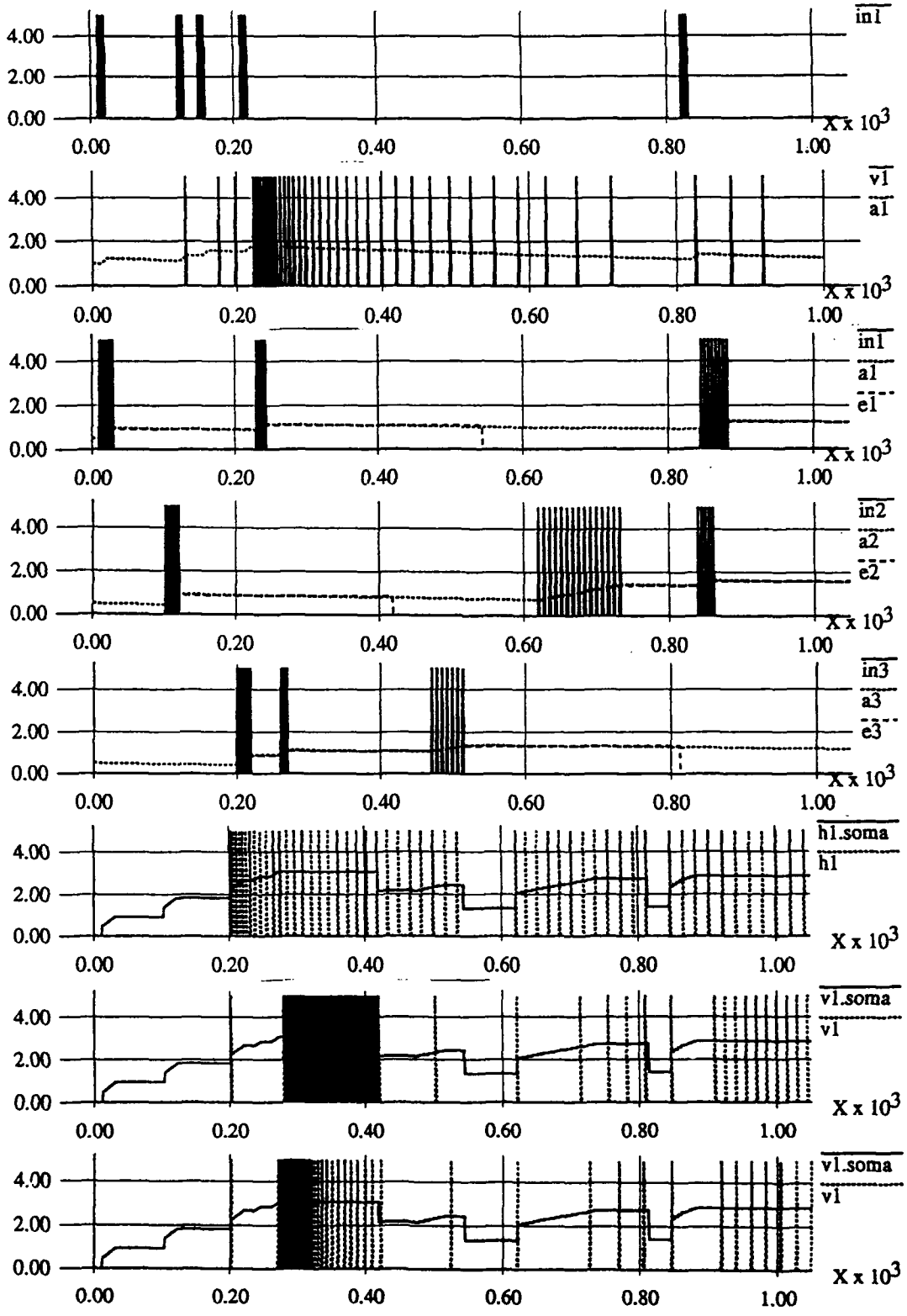


**Fig. 21:** Spatiotemporal Encodings Produced by HTPE Simple Cells.

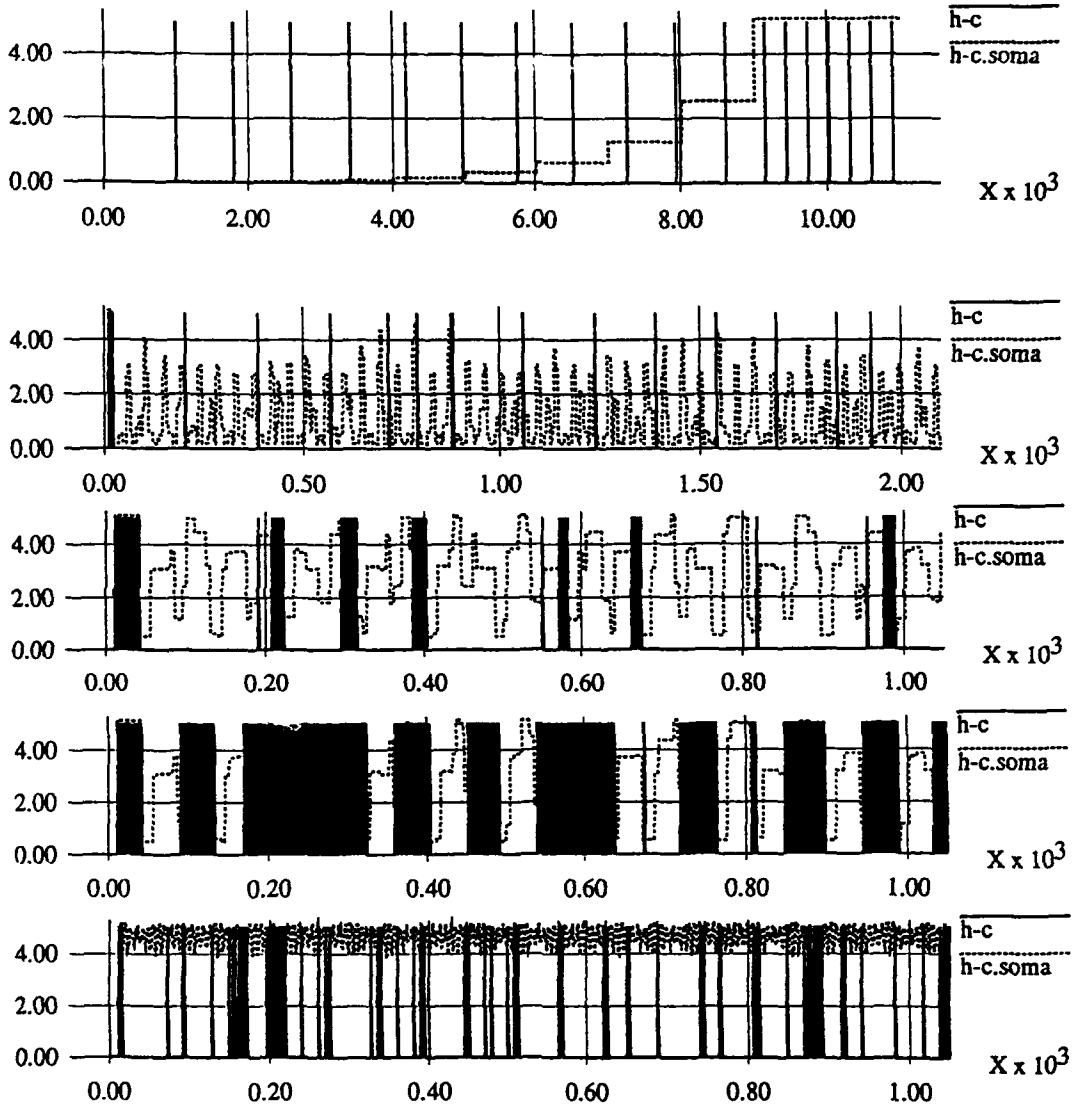


**Fig. 22:** Illustration of Depth-to-Disparity Mapping.

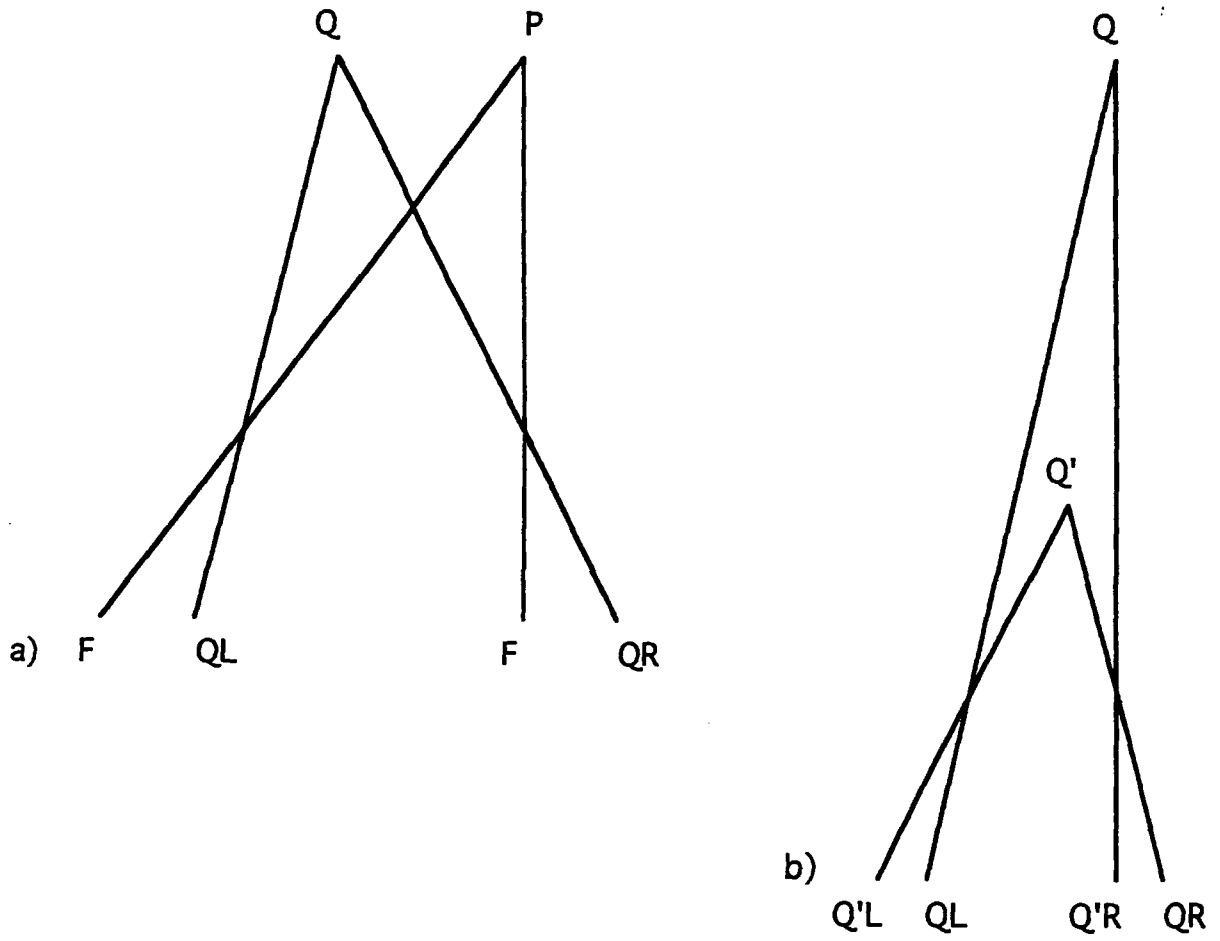


Fig. 23: Block Diagram of IRIS Integrated Active Vision System.

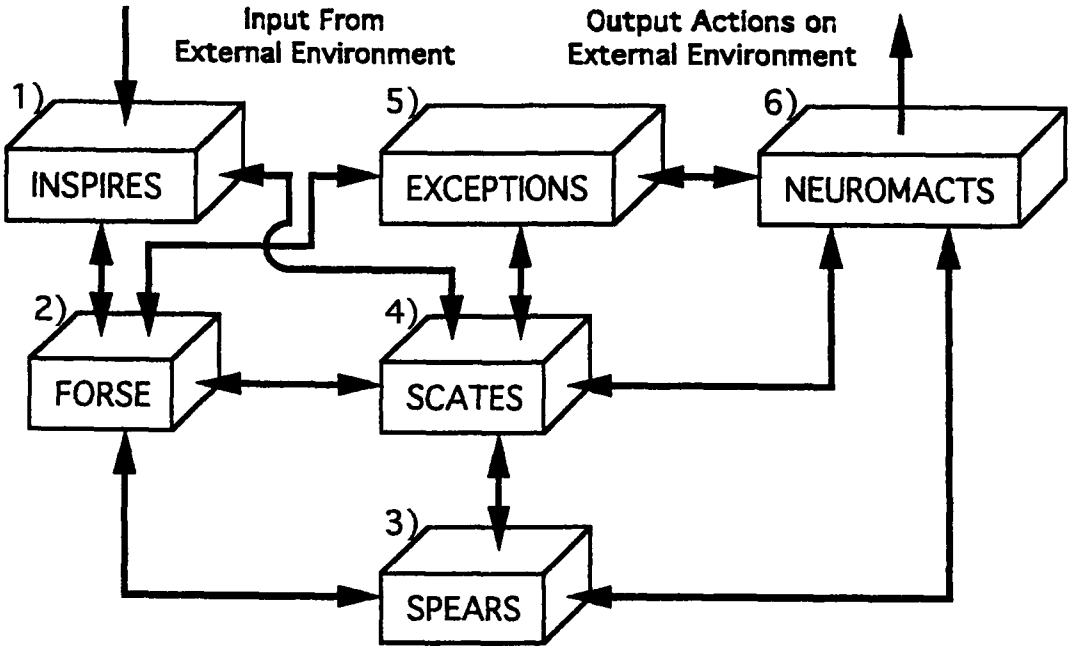


**Fig. 24:** Block Diagram of Feature and Object Recognition System Configurations.

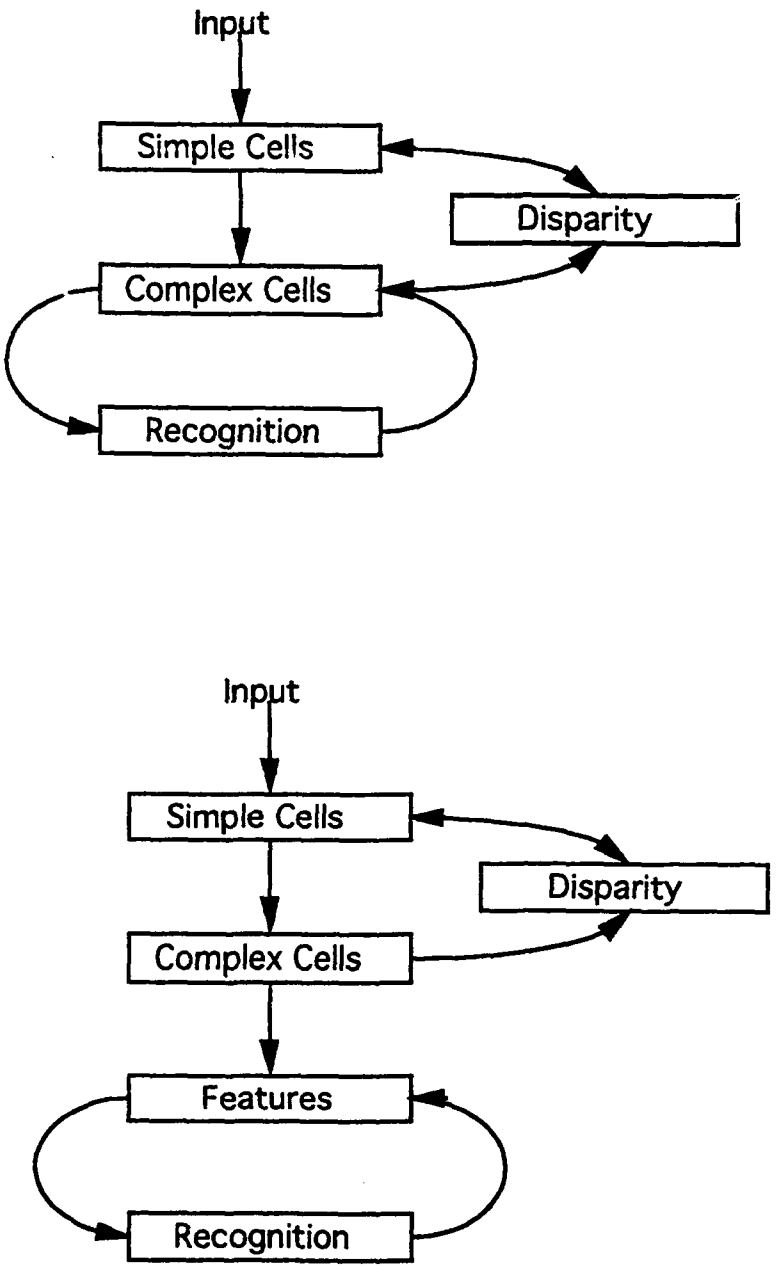


**Fig. 25:** Hierarchical Nature of the Grammar-Based Approach.

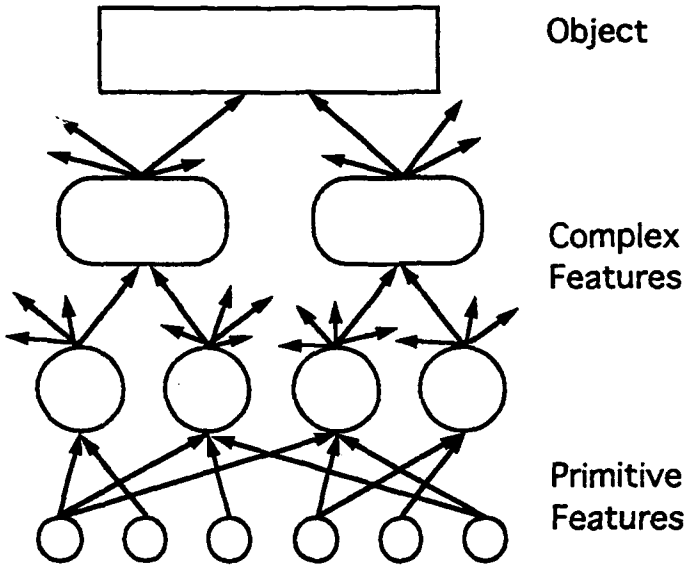


**Fig. 26:** EPSP, IPSP, SPSP, and AP Waveforms Produced by HTPE.