

1995-118834

N95-25254

P-10

Missileborne Artificial Vision System (MAVIS)

David K. Andes, James C. Witham, Michael D. Miles

Naval Air Warfare Center - Weapons Division

China Lake, CA 93555

ABSTRACT

Several years ago when INTEL and China Lake designed the ETANN chip, analog VLSI appeared to be the only way to do high density neural computing. In the last five years, however, digital parallel processing chips capable of performing neural computation functions have evolved to the point of rough equality with analog chips in system level computational density. The Naval Air Warfare Center, China Lake has developed a real time, hardware and software system designed to implement and evaluate biologically inspired retinal and cortical models.

The hardware is based on the Adaptive Solutions Inc. massively parallel CNAPS system COHO boards. Each COHO board is a standard size 6U VME card featuring 256 fixed point, RISC processors running at 20 MHz in a SIMD configuration. Each COHO board has a Companion board built to support a real time VSB interface to an imaging seeker, a NTSC camera and to other COHO boards. The system is designed to have multiple SIMD machines each performing different Corticomorphic functions.

The system level software has been developed which allows a high level description of Corticomorphic structures to be translated into the native microcode of the CNAPS chips. Corticomorphic structures are those neural structures with a form similar to that of the retina, the lateral geniculate nucleus or the visual cortex.

This real time hardware system is designed to be shrunk into a volume compatible with air launched tactical missiles. Initial versions of the software and hardware have been completed and are in the early stages of integration with a missile seeker.

INTRODUCTION

The onboard processing requirements of air intercept missiles are some of the most demanding imaginable. This is especially true for missiles with imaging focal plane array detectors. Input is measured in megabytes per second. The volume available is a few cubic inches. Decisions are required in milliseconds. The power available is just a few watts and heat dissipation is minimal. Then the system must live in an environment that includes salt air, desert heat, Arctic conditions, high humidity and rapid altitude changes. Aircraft systems have similar constraints but the power, volume and heat dissipation problems are slightly less severe. If we are to survive in a competitive world, however, we must continue to upgrade the internal intelligence of our systems.

Biological systems have met and overcome even greater competitive challenges in real-time embedded computing. Biosystems have similar constraints in power, volume, heat dissipation while requiring high speed computation including high data rate sensors of several varieties. There should be much to learn from the many, highly successful, integrated, real-time biocomputers that surround us every day. The MAVIS project is an attempt to do just that.

Biological Computation Systems

The following is a partial list of some of the salient characteristics of biological computation systems:

1. *Massive parallelism* is the first obvious characteristic. We cannot hope to come even close to the biosystems in this area but at least it gives a definite direction in which to move. Many simple processors working almost independently can clearly achieve great results.

2. Most biocomputation is based only on *locally available information*. Transmitting information beyond a few tenths of a millimeter becomes very expensive.
3. There is a *lack of emphasis on precision* in the elementary processors (neurons). In the cases where more precision is necessary more elementary processors are dedicated to the task.
4. Local computational centers share information with *several* other local centers in a *bi-directional* manner. Computation is shared in a non-hierarchical or only a semi-hierarchical manner. In fact most of the information entering the local processing centers is not raw sensor data but partially processed information from other local centers.
5. The computational components of biosystems are *finely tuned* parts of a whole system. Competition has not allowed much that is inefficient or unnecessary. The processing devoted to sensor data is well matched to the quality and importance of the information.

Corticomorphic Processing

The mammalian vision system has some special structural characteristics which are clearly specialized for the processing of two dimensional image information. An abstraction of the form of this system is used in the MAVIS project and has been given the name Corticomorphic Processing. Although this model is an abstraction of the processing centers of the visual system (such as the retina and patches of visual cortex) it is hoped that models of other areas of the cortex will fit into this general form. The Corticomorphic abstraction is an Artificial Neural Network (ANN) though not of one of the standard forms (e.g. Backpropagation, ART, Hopfield, etc.).

The early processing stages of the visual system (areas like the retina, the Lateral Geniculate Nucleus, primary visual cortex, V2, V3, etc.) have computational forms which are similar. Each area is a "patch" of computational elements laid out in a form which preserves, at least locally, the two dimensional relationships in the original image. Within each of the patches there are various types of neurons arranged in sheets or layers that run throughout the entire patch. Even though the neurons on different sheets perform very different functions the rough topology of the original image is preserved in each sheet. A column cut vertically into a patch through all the sheets will find neurons which only respond to a small local area of the original image. Inputs into each sheet of a patch come in through topology preserving maps from other sheets. Most inputs into a sheet are from sheets within the same patch but some come from sheets within other patches. The strengths of the interactions between neural processing elements can be approximated by the mathematical form of convolution kernels. This is an approximation that is only locally true in real biosystems since it requires exactly the same processing to take place throughout the entire length and width of a patch.

Formalism

The introduction of some formalism may make all this more precise if not clearer. Let

$$O(x,y,i,j,t)$$

be the output value of the neural processing element at the (x,y) position of the image space in the i-th layer of the j-th patch at time t. Then

$$L(m,n) = \{ O(x,y,i,j,t) \} \text{ for } i=m \text{ and } j=n$$

is the m-th sheet or layer in the j-th patch. Note that L(m,n) is a set of neural processing elements. Note also that we have shifted from the more descriptive word "sheet" to the more traditional ANN term "layer". Then let

$$P(i) = \{ L(m,k) \} \text{ } k=j$$

be the i-th patch. Note that P(i) is a set of layers.

Typically the number of layers in a patch runs from three to ten and only a few of the layers in a patch have outputs to layers in other patches. The output value of the neural processing elements of a layer $L(i,j)$ is calculated as follows:

$$O(x,y,i,j,t) = F_{ij} \left(\sum (a_{ij,s,p} + g_{ij,s,p} \sum k_{ij,s,p}(l,m) O(x-l,y-k,s,p,t-b_{ij,s,p})) \right) \quad (1)$$

The first sum is a sum over s and p where p runs over all patches driving this layer $L(i,j)$ and s runs over all layers in p which connect to the layer $L(i,j)$. The second sum is also a double sum over l and m which run through enough positive and negative integers to cover the kernel $k_{ij,s,p}$.

In this expression:

F_{ij} is the nonlinear function associated with the neural processing elements of the layer $L(i,j)$.

$k_{ij,s,p}$ is the kernel weight function which determines the effect of the $L(s,p)$ layer on the $L(i,j)$ layer.

$b_{ij,s,p}$ is either zero (no time delay) or one (one time step delay) depending on whether the information affecting $L(i,j)$ from $L(s,p)$ is to be current or delayed.

$a_{ij,s,p}$ and $g_{ij,s,p}$ are appropriate offset and gain numbers affecting the action of layer $L(s,p)$ on layer $L(i,j)$.

In plain English this amounts to the following: each layer in each patch is calculated by applying a set of kernel convolutions to one or more other layers, summing the results and then passing it through a possibly non-linear function. Gains, offsets and time delays may be applied where necessary.

Although the sums look complex they typically contain only one to three kernel interactions with most of the interactions occurring within the same patch (i.e. $j=p$). In fact a layer may interact with itself in which case $j=p$ and $i=s$ and $b_{ij,s,p}$ must be one. This self interaction allows for temporal integration (both point and area).

One more basic construct is useful and that is the idea of a column. Let

$$C(u,v,p)$$

be the symbol for the column centered on the point (u,v) in image space on patch p . Then if

$$R_x(C) \text{ and } R_y(C)$$

are the x and y radii of the column we have

$$C(u,v,p) = \{ O(x,y,u,v,t) \in L(i,j) \text{ such that } |x-u| < R_x(C) \text{ and } |y-v| < R_y(C) \} \quad (2)$$

That is a column is the set of all points (outputs of neural processing elements) in pieces of sheets (or layers) from a single patch which are all cut to the same size and all of which are centered at the same place in image space. Note that for $C(u,v,p)$ the values of $u, v, R_x(C), R_y(C)$ need not be integers.

History of Embedded Neurocomputing at China Lake

For the past fifteen years the Office of Naval Research has been funding work at China Lake with the aim of increasing the capability of embedded computational systems for air intercept weapons. Most of the work described in this paper was done under this ONR funding although a significant portion of the early work in several of the areas was started under local funding at China Lake.

In the early 1980's it became clear that traditional Artificial Intelligence techniques had only limited utility for embedded real-time systems in air intercept missiles. This was due mostly to the inability of the hardware of the time to match the severe constraints imposed by these systems. In the mid 1980's the

biologically inspired field of Artificial Neural Networks showed promise of helping to overcome this computational bottleneck. The ideas were amenable to implementation in high speed, parallel, analog circuitry and learning algorithms could be used to circumvent the problems associated with analog imprecision. Early experiments and designs at China Lake led to the development of the Intel ETANN chip [1]. This chip is capable of about three billion operations per second in a fraction of a square inch.

In 1989 the Missileborne Artificial Neural Network Demonstration (MINND) program was initiated to exploit the availability of the new computational power. The MINND program was successfully completed in 1992 with real time demonstrations on real air targets [2]. The architecture of the MINND computer allowed a simple version of the Corticomorphic Processing scheme to be implemented. The fixed form of the analog circuitry, however, put rigid constraints on the types of computations that could be performed. Toward the end of the MINND program it became clear that digital computation was catching up to the analog when total system level computational density was considered. In particular the Adaptive Solutions CNAPS chip [3] had characteristics that allowed us to design the current MAVIS system. MAVIS has system level performance similar to the ETANN based MINND system but without the associated analog problems. Packaging techniques are available which allow the design of the MAVIS system to be reduced enough to fit the constraints of an air intercept missile. The sections of this paper that follow describe the hardware and software components of the MAVIS system.

MAVIS HARDWARE OVERVIEW

The MAVIS system is built around the Adaptive Solutions CNAPS chip. Each chip has 64 fixed point, RISC processors that currently operate at 20 MHz. These processors are designed to operate in an SIMD configuration where several CNAPS chips may be under the control of a single sequencer chip [4]. Each of the 64 processing nodes (PNs) on each CNAPS chip has an adder, a multiplier, a logic unit, 4K bytes of local memory, several general purpose registers, and inter-PN bussing. The system uses the Adaptive Solutions COHO boards [5] each of which mounts four CNAPS chips for a total of 256 PNs per board. The MAVIS system is designed to accommodate several of these COHO boards each of which is used to implement one patch of Corticomorphic processing. A high speed bus intercommunication scheme has been designed to allow high bandwidth injection of sensor data as well as high bandwidth inter-patch communication.

An overview of the initial MAVIS system can be seen in Figure 1. It shows an imaging seeker connected to the MAVIS card cage, a Motorola MVME-147 board (68030 processor), two Adaptive Solutions Inc. COHO boards, two NAWC designed COHO Companion boards, and a NAWC designed Custom I/O board. The diagram also shows two video display monitors and two VCRs used for displaying and recording raw and processed video.

Adaptive Solutions Inc. has a set of integrated tools that can be used to develop and debug code for their COHO board by using a SUN SPARC station connected to the MVME-147 via an ethernet network. Code is developed and compiled on the SUN workstation and then downloaded to the COHO board to run.

Hardware Specifics

COHO Board

The COHO board is a commercially available 6U VME board. The major components of the board are highlighted in Figure 2.

The board has provisions for attaching peripheral devices or memory onto its local bus. The name of this local bus is the CNAPS/VME local bus (CVLB). The CVLB is an implementation of the company's ADAPTbus™ applied to this specific board and its peripherals. There is a 100 pin impedance matched connector on the COHO board which provides access to the CVLB. It is this connector that the COHO board uses to interface to the COHO Companion board.

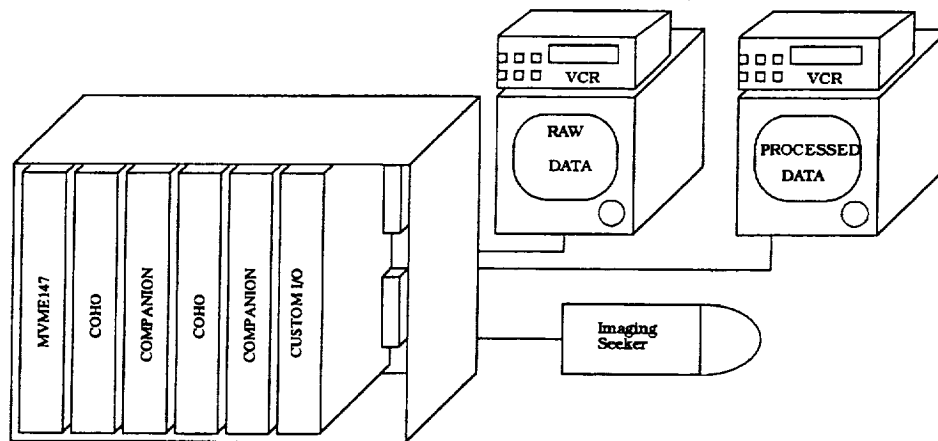


Figure 1

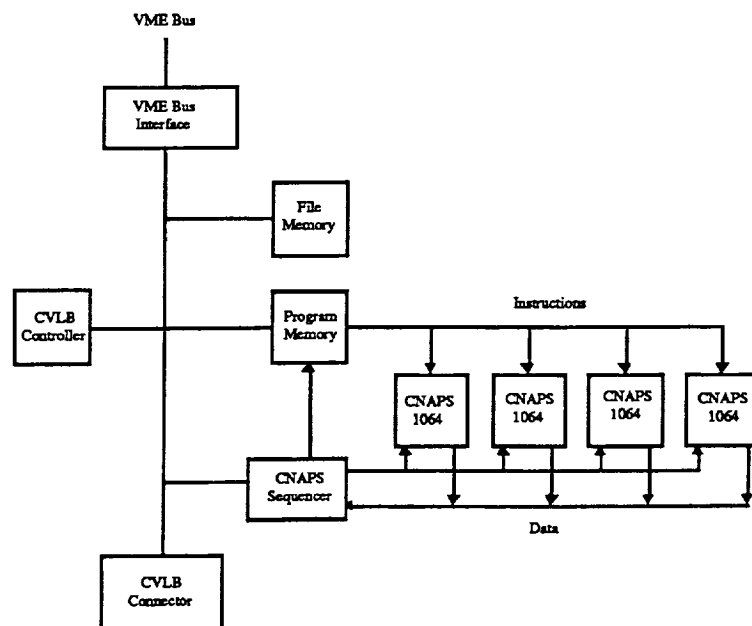


Figure 2

COHO Companion Board

A block diagram for the COHO Companion board is shown in Figure 3. This architecture, made up of two ping-pong memories, was chosen because it allowed images to be read from or written to both memories simultaneously. For instance, as an incoming image is being written into Bank 1, an image can be read out of Bank 2, processed and then written back to Bank 2 without impeding the incoming image. When both tasks are finished the memories are swapped, so that the image in Bank 1 may be processed while a new incoming image may be written into Bank 2.

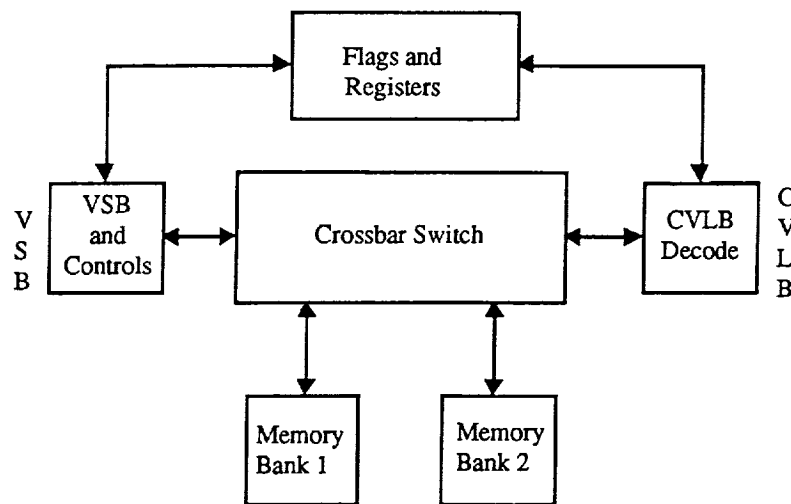


Figure 3.

If one assumes an image patch of 128 by 128 and a frame rate of 60 frames per second the amount of data that is actually passed into the system is approximately 1 MByte per second. With the MAVIS system setup, data is processed on each COHO board (patch) and is available for display only when sent over an interconnection bus. Thus under these assumptions with only a single COHO/COHO Companion board pair the final I/O requirements are only about 2 MBytes/sec. When more than a single pair of boards are used, however, there will be interaction between boards and, with more interaction, more bus bandwidth is required. If larger images or higher video rates are required the bus bandwidth also increases. For these reasons, it was decided to offload the data from the VME bus and use the VSB bus (VME Subsystem Bus). The current implementation is able to move data at 12 MBytes/second over the VSB. Figure 4 shows the buses and the type of data that is transferred on each bus.

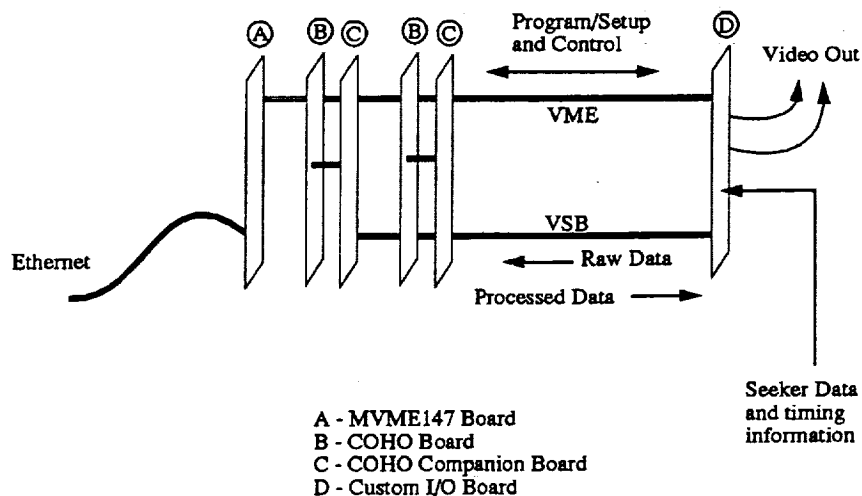


Figure 4

Custom I/O Board

The Custom I/O board was fabricated to comply with the digital video and timing signals for an imaging seeker. The board is also capable of displaying the incoming digital video, plus an extra video channel that

may be used to show the results of processed or intermediate data. It is also capable of selecting an Area Of Interest (AOI) of variable size and location, from the incoming video, and transmitting it on the VSB Bus.

As shown in Figure 5 the system is based around a pair of dual ported memories, one for the input, and one for the output. The output video frame's timing is in lock step with the input video frame's timing. This feature could be used to reinsert the processed digital video back into the data stream that it was taken from.

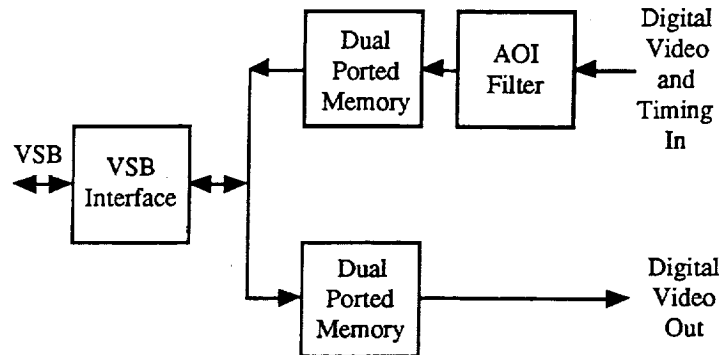


Figure 5

System Options

Having the MAVIS system tied directly to a real missile seeker has many advantages for answering questions related directly to that particular system. There are, however, many disadvantages associated with such a system. A second system option is also being implemented which is much more general than the single seeker system described above. The second system uses a pan/tilt unit with a camera mounted to it in place of the imaging seeker. Several additional boards are required to interface to a camera with a pan/tilt unit: a frame grabber/display board, a D/A (Digital to Analog) board, and a single board computer (SBC). A general purpose microprocessor on the SBC receives information from the COHO board with a target location and generates the angle rates for the pan/tilt unit and sends them out via the D/A board. The microprocessor can also take slave commands from a joystick for external target designation.

MAVIS SOFTWARE OVERVIEW

The system level software is designed to combine flexibility with ease of use in the implementation of a variety of Corticomorphic structures. The system level software is written in C and takes a text file containing Corticomorphic descriptors and produces microcode which is native to the CNAPS processors.

The first step in implementing a Corticomorphic concept is to develop a block diagram of the system to be modeled. Figure 6 shows a relatively simple model of the outer retina. The model itself is broken up into several layers. These layers themselves are idealized models of distinct types of retinal neurons. The boxes labeled with the capital letter K and a number refer to the kernel which will be used in the convolutional interaction between the layers. A kernel is a square matrix made up of integer weights designed to have a specific effect, such as edge enhancement or smoothing.

As shown in equation (1) the creation of each layer is dependent upon several things: the other layers in the model, the kernels with which the layers will be convolved, and the method of combining the results. The software allows for simple definitions of feedback paths both from a layer further along in the model path and from a layer to itself. This self interaction is accomplished by storing a layer in memory when it is created at time $t-1$, so that it may be used in the creation of a layer at time t .

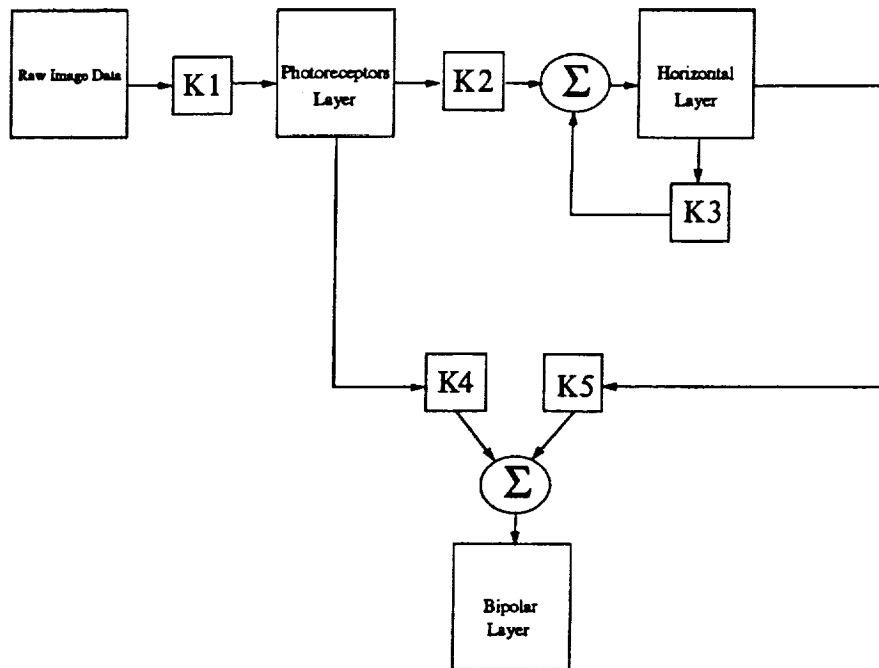


Figure 6

From the block diagram, the user must create a model file and kernel files. A model file is a simple text file containing a description of the elements the user wishes to include in the model. Kernel files are text files containing the dimensions, weights, gains and offsets for a kernel. The system software reads the model file, which references the kernel files as they are needed and uses its' specifications to generate another file containing CNAPS microcode. This microcode is assembled using the CNAPS assembler and then loaded into the COHO program memory space. At this point, the user needs only to assert a start command for the software to assume command of the hardware system.

There are certain details the software must accommodate to implement equation (1). Figure 7 shows the application of a kernel $(k_{i,j,s,p})$ to the intersection of a layer $L(i,j)$ and a column $C(u,v,p)$ as described in equation (2). The pixels surrounding this portion of the column are part of a software construct known as a tile border. As indicated in the figure, the tile border and the column section comprise the tile itself. In order for the kernel to be applied so that the result has the proper correspondence to the pixels along the edges of the column, extra information is required. This extra information is borrowed from neighboring PN's and comprises the tile border. If no tile border was constructed, and the kernel was simply applied as in Figure 8, the result would be the shrinking of the column size as in Figure 9.

The patches referred to in the equations are actually separate COHO boards. The software allows the user to specify which board will act as which patch and which layers the patch will be responsible for processing.

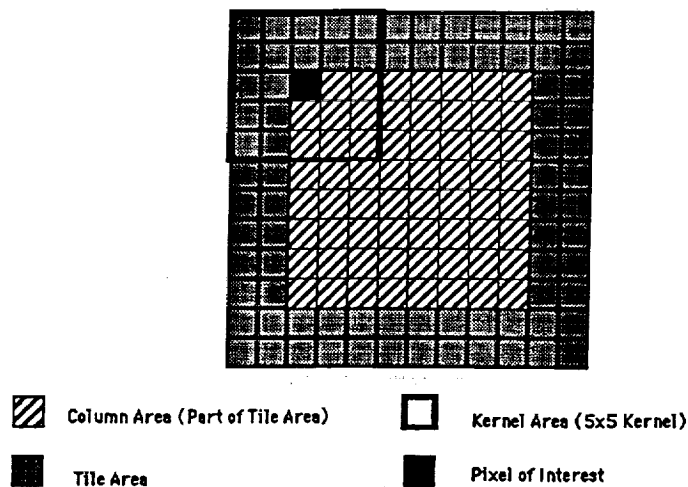


Figure 7

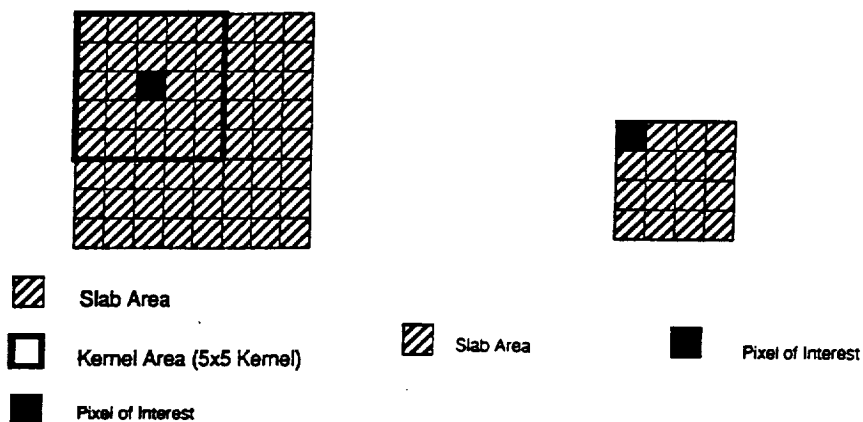


Figure 8

Figure 9

GENERAL NOTES

There are several extensions to the basic Corticomorphic structure which are already planned. None of these require a modification to the form of the hardware.

1. The simplified computational form of equation (1) can be extended to allow the multiplication of convolutions of layers as well as the sum. Sums and products could also be mixed in the same evaluation. This modification has already been tried and is not included in equation (1) mainly because it complicates the formalism and the write-up. Multiplication takes no more time than addition and hence this modification costs nothing in compute time. The same cannot be said of the next two extensions.
2. The terms in the equation (1) which appear as constants (such as kernel weights, gains and offsets) could be made to vary with time since they are stored in memory local to each controller.
3. Time delays of longer than one frame have been implemented. The cost is in local memory and some in compute time.

It is important to note that most of the current image processing schemes (neural net or otherwise) can be put into the form of equation (1) or a minor extension of it as given above. Hence the MAVIS system provides a good real-time test bed for many current image processing ideas.

CONCLUSION

MAVIS is an attempt to produce a computational structure which emulates the form of the processing used in the mammalian vision systems. The eye and the brain are a coupled system which obtains an understanding of the environment by interacting with it. It is hoped that the investigation of this complex interaction will shed light on the functioning of real cortex as well as allowing us to design better sensing systems for both military and non-military applications.

REFERENCES

- [1] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10,240 "floating gate" synapses," in Proc. Int. Joint Conf. Neural Networks (Washington D.C.), 18-22 June 1989, p. II-191.
- [2] M. Mumord, D. Andes, and L. Kern, "The Mod 2 Neurocomputer System Design," in IEEE Transactions on Neural Networks, Vol 3., No. 3, May 1992.
- [3] D. Hammerstrom, E. Means, M. Griffin, G. Tahara, K. Knorpp, R. Pinkham, and B. Riley, "An 11 Million Transistor Digital Neural Network Execution Engine," in IEEE International Solid-State Circuits Conference, 1991, p. 180-181.
- [4] D. Mueller, and D. Hammerstrom, "A Neural Network Component," in IEEE International Conference on Neural Networks (San Francisco), March 1993.
- [5] T. Skinner, "Digital Signal Processing on a Multiprocessor System," Electronic Design, Penton Publishing, Cleveland, Ohio, 7 February 1994.