A Parallel Implementation of Collective Learning Systems Theory: Adaptive Learning Image Analysis System (ALIAS)

Professor Peter Bock

The University of Ulm (visiting)
Research Institute for Applied Knowledge Processing (FAW)
Ulm, West Germany

Department of Electrical Engineering and Computer Science
The George Washington University
Washington DC USA

with GWU Graduate Students Richard Rovner, C. Joseph Kocinski, Hilary Holz, and Glenn Becker

ABSTRACT

An alternative to preprogrammed rule-based Artificial Intelligence is a hierarchical network of cellular automata which acquire their knowledge through learning based on a series of trial-and-error interactions with an evaluating Environment, much as humans do. The input to the hierarchical network is provided by a set of sensors which perceive the external world. Based upon this perceived information and past experience (memory), the learning automata synthesize collections of trial responses. Periodically the automata estimate the effectiveness of these collections using either internal evaluations (unsupervised learning) or external evaluations from the Environment (supervised learning), modifying their memories accordingly. Known as Collective Learning Systems Theory, this paradigm has been applied to many sophisticated gaming problems, demonstrating robust learning and dynamic adaptivity.

Based on a versatile architecture for massively parallel networks of processors for Collective Learning Systems, a Transputer-based parallel-processing image processing engine comprising 32 learning cells and 32 non-learning cells has been applied to a sophisticated image processing task: the scale-invariant and translation-invariant detection of anomalous features in otherwise "normal" images. In cooperation with Robert Bosch GmbH, this engine is currently being constructed and tested under the direction of the author at the Research Institute for Applied Knowledge Processing (FAW-Ulm) as Project ALIAS: Adaptive Learning Image Analysis System. Initial results indicate excellent detection, discrimination, and localization of anomalies.

INTRODUCTION TO COLLECTIVE LEARNING SYSTEMS THEORY

Based on the seminal work in learning automata theory by S. Lakshmivarahan [LAK81] and K.S. Narendra [NAR77], the Collective Learning paradigm was first proposed in 1976 by the author [BOCK76]. A group of Collective Learning Automata (CLA) embedded in an external evaluating Environment is known as a Collective Learning System (CLS), as diagramed in Figure 1. The memory of each CLA is a state transition matrix (STM) comprising a set of probability (or weight) vectors which govern the transitions from its stimulus domain to its response range. At the outset of learning each CLA is totally ignorant; all legal responses for each stimulus are initialized as equally likely. [BWW85]

Each CLA receives a continuous stream of stimuli from its Environment or other CLAs in its group. Based upon a stimulus, each CLA applies a selection function to synthesize a response based on the probability (or weight) vector in its memory corresponding to the stimulus. This response may either be submitted to the Environment, modifying the state of the Environment and (possibly) causing it to generate a new stimulus for the group of CLAs; or the response may be submitted as a new stimulus directly to other CLAs in the group. Periodically the Environment issues an evaluation of the collection of responses it has received to one or more of the CLAs in the group. Based on this evaluation, each CLA computes a compensation and updates its memory accordingly. This "Algedonic Cycle" is repeated until the CLA consistently synthesizes "correct" responses, i.e., responses which elicit an acceptably high positive evaluation from the Environment. The stimulus domains and response ranges of an STM may be preprogrammed (genetic) and/or dynamically acquired (self-organizing). Groups of CLAs may be linked together in hierarchical networks to form larger "modules" which can convolve the simple responses synthesized by their component groups into more complex responses.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1990 ACM 089791-348-5/90/0002/0457 $1.50

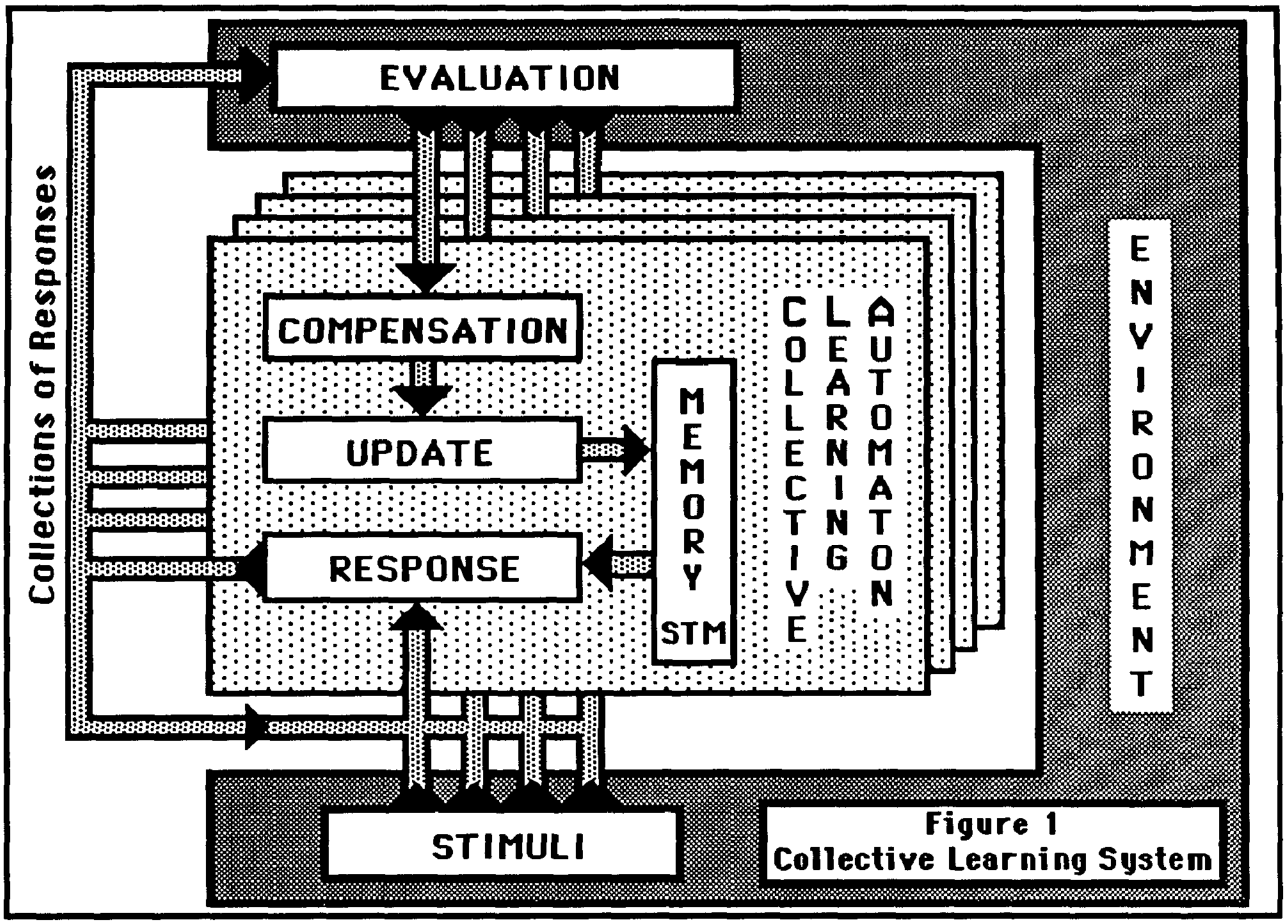


Figure 1
Collective Learning System

As shown in Figure 2, the process of receiving a stimulus, synthesizing a collection of responses, evaluating this collection, and applying a compensation and updating the memory constitutes a "stage" of learning. The CLA uses the evaluation to modify the elements of its memory corresponding to each stimulus/response pair (interaction) used during the stage (history). The compensation function serves to temper the evaluation based on a number of factors, such as the maturity of the CLA (number of stages) and the degree of saturation of the interaction probabilities (or weights). Using this compensation, the update function then changes the CLA's memory for each interaction in the history of the stage. The update function may be based on several factors, such as the saturation of the interaction probabilities (or weights), and the order of the interactions.

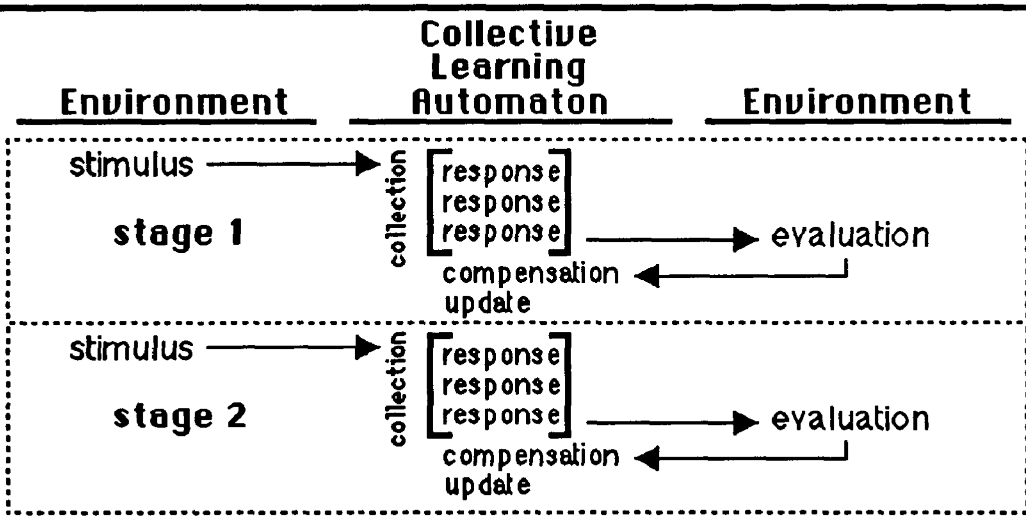


Figure 2 - Scenario of a Collective Learning Automaton

The memory of a CLA is a state transition matrix (STM). The domain and range of this transition space are the set of all the stimuli which the CLA can currently recognize (stimulants) and the set of all possible legal responses to these stimuli (respondents), respectively. The elements of the matrix may be either probabilities or weights. For a given stimulus, the values of the elements of the corresponding column vector in the STM govern the selection of the CLA's response. A typical and useful selection function simply selects the response with the highest probability (or weight), arbitrating ties randomly.

For many tasks the enumeration of all possible stimulus/response transitions can be prohibitively large. However, the useful subsets of the domains and ranges often converge to a very small subset of the complete combinatorial set, with the vast majority of the stimulus/response transitions proving (through learning) to be strategically unimportant. Dynamic allocation and deallocation of memory and the use of efficient data structures can often forestall combinatorial state-space explosions.

Cell Structure

A CLA is instantiated as an entity known as a *learning cell,* whose structure is shown in Figure 3 [BWW85]. To be quite general, a cell may be defined as either learning or non-learning. A non-learning cell has no memory; the selection function is simply an algorithm that transforms stimuli into responses in a preprogrammed manner. A learning cell, on the other hand, acquires its strategy through experience, and is therefore of much greater interest in the study of sophisticated collective learning systems.

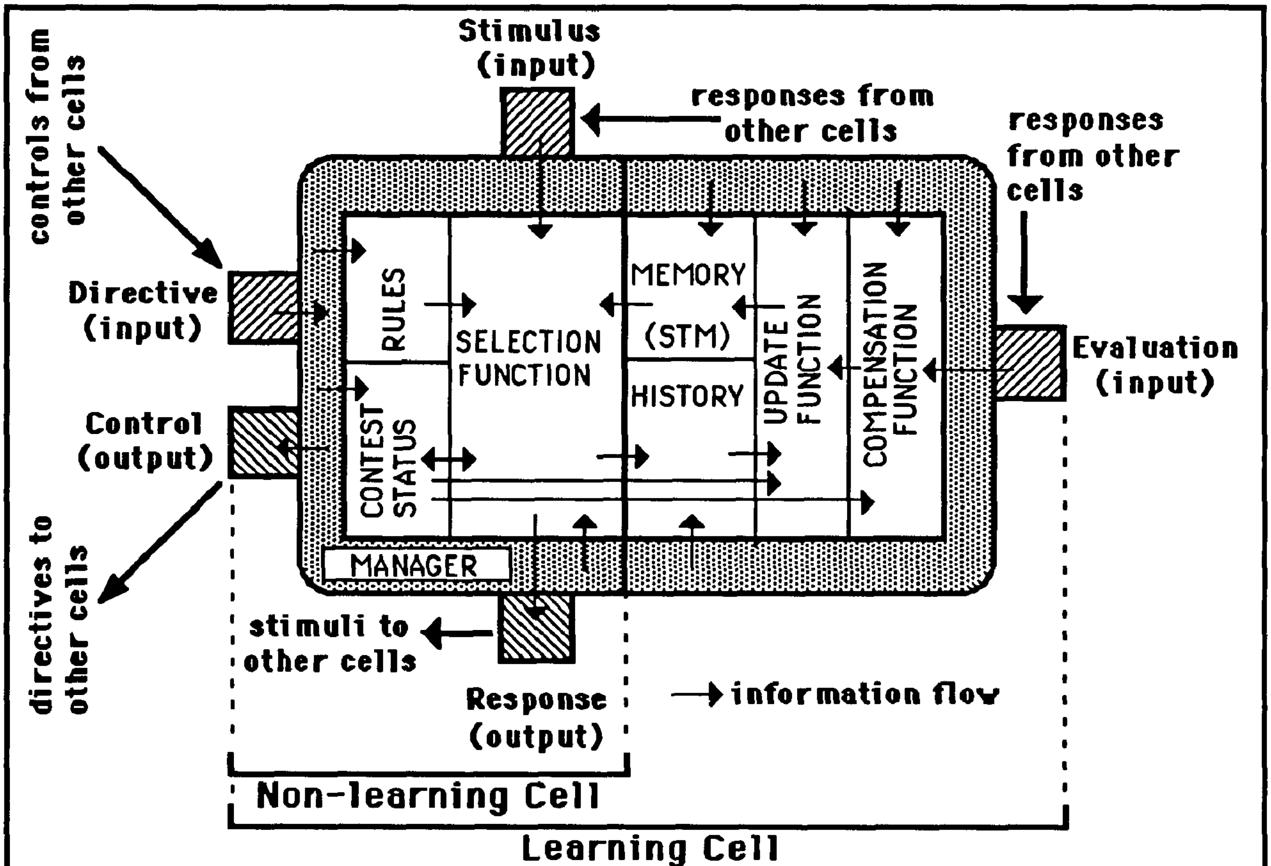


Figure 3 - Cell Architecture

A learning cell has three inputs (stimuli, controls, and evaluations) and two outputs (responses and directives). A stimulus triggers the selection function which selects a response based on the cell's memory (STM), the state of the Environment (contest status), response constraints (rules), and the value of the stimulus itself. Evaluations elicit no response, but rather cause a modification to the cell's strategy as encoded in its memory. Based on the evaluation, a compensation is generated, and the appropriate elements of the memory are updated based upon this compensation and the history of interactions during the stage which has just ended.

Controls (inputs) and directives (outputs) are used by cells connected in networks to exchange information other than stimuli, responses, and evaluations. They may be used, for example, to transfer part or all of one cell's memory to another cell, to modify a cell's functions (e.g., selection, compensation, and update), and/or to allocate and deallocate cells from the network. Such communication is termed "genetic", and the process of modification of the cell's structure or functions is termed "self-organization".

Individual cells may be embedded in a hierarchical network to accomplish more sophisticated cooperative and competitive tasks. In addition to stimuli from the input stream, cells may receive stimuli via feedback from other cells (or themselves) to facilitate local and global consolidation of learned strategies (homeostasis). Evaluations which govern supervised learning are received directly from an external Environment. The network interfaces for a learning cell network are shown in Figure 4.

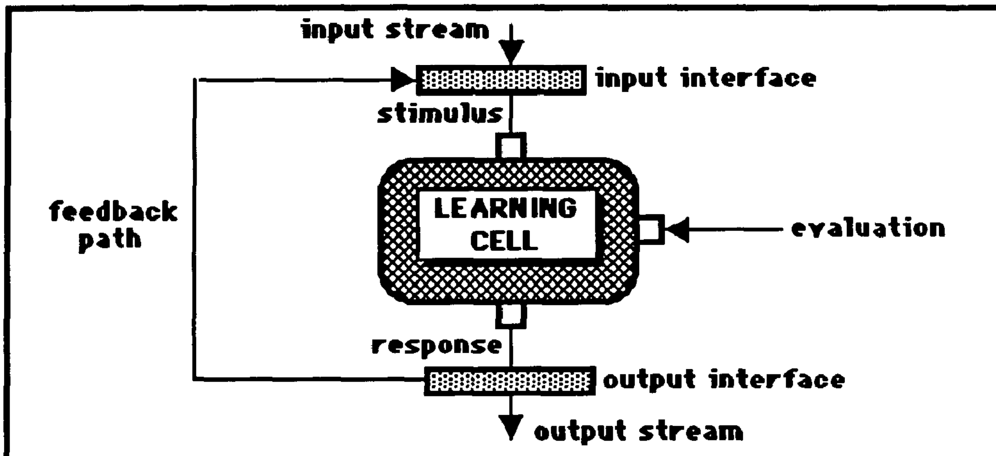


Figure 4 - Learning Cell Network Interfaces

The input interface transmits a stimulus packet to the cell, originating either from the input stream or from the feedback path. *If the cell is not busy and the stimulus is recognized as a member of the cell's stimulus domain, it generates a response. If the cell is* busy, the stimulus is missed. (Missed stimuli could be buffered, but this is a degenerate solution since any buffer must necessarily be of finite size, and the number of stimuli arriving while the cell is busy could conceivably become greater than any specified buffer size.) If the stimulus is not recognized, either the stimulus may be ignored or the cell may generate a special response indicating lack of recognition. The output interface directs the cell's responses to the output stream and the feedback path. If needed, a number of successive responses may be assembled into packets in the output interface before being sent to the output stream and feedback path.

Network Architecture

To permit the development of an arbitrarily complex hierarchical communication network for embedding large numbers of cells, the input and output interfaces of a cell are each connected to a network with a ring topology, as shown in Figure 5. It is the presence of a feedback path connecting the output ring to the input ring which defines this assembly as a "module". [BOCK288]

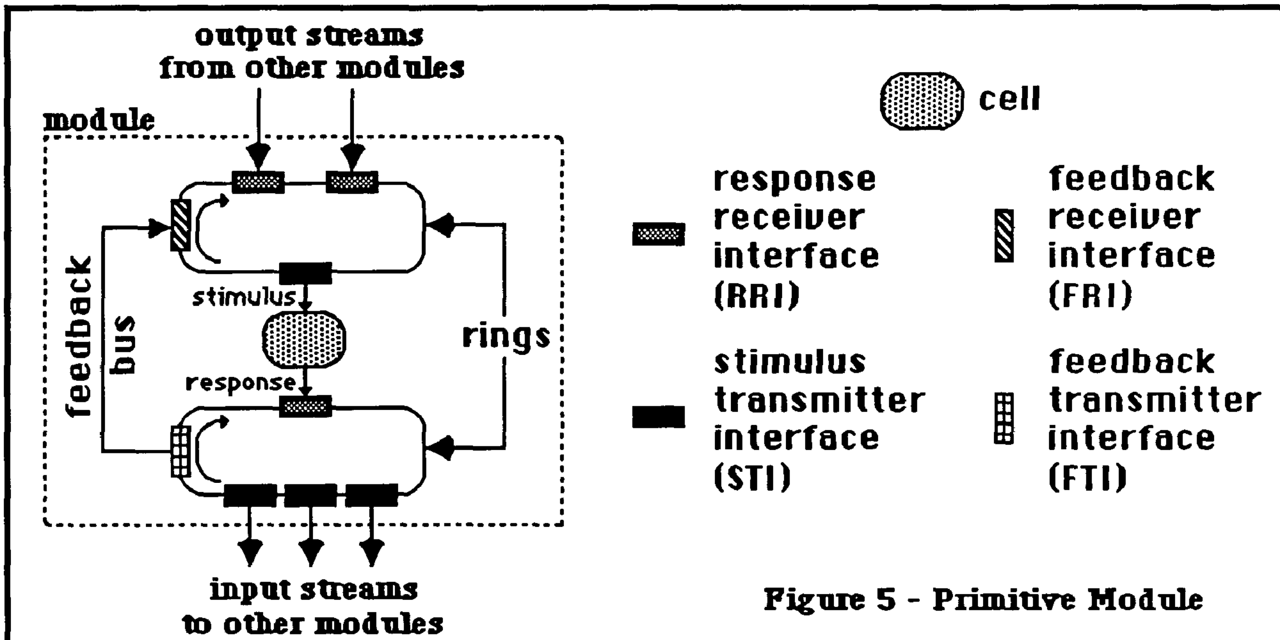


Figure 5 - Primitive Module

A dynamic slotted-insertion protocol is specified for the input and output rings to allow efficient synchronous transmission of variable length packets, since the lengths of stimuli and responses may vary, and the timing of messages may be critical. Traffic permitting, a ring topology (rather than, say, a bus topology) allows data packets (stimuli or responses) to circulate indefinitely on the communication path. This provides an additional memory mechanism for the network by allowing ignored or missed packets to be reconsidered during subsequent circuits.

The primitive module is the building block of a hierarchical network of modules. While the primitive module consists only of cells, the components of higher level modules may themselves be modules, defined as before by the presence of a feedback path from the output ring to the input ring. Any module, including primitive modules, may be extended horizontally by placing modules or cells in parallel on the same ring, and/or vertically by linking rings serially in succeeding layers, as shown in Figure 6. In addition, selector cells may be introduced between a stimulus transmitter interface and a downstream module. This allows a set of responses from an entire module to be pruned, forwarding only a subset of them as stimuli for the downstream module. Moreover, selector cells may also be used to transform several incoming responses into stimuli whose format is quite different from that of the incoming responses. Selector cells may be learning or non-learning. The cells in the primitive modules synthesize responses to the stimuli which have been passed to them via the rings down through the hierarchy. The responses of these cells serve as the stimuli to cells at the next layer in the same primitive module, or, via ascent and descent through the hierarchy, to cells in other primitive modules elsewhere in the network.

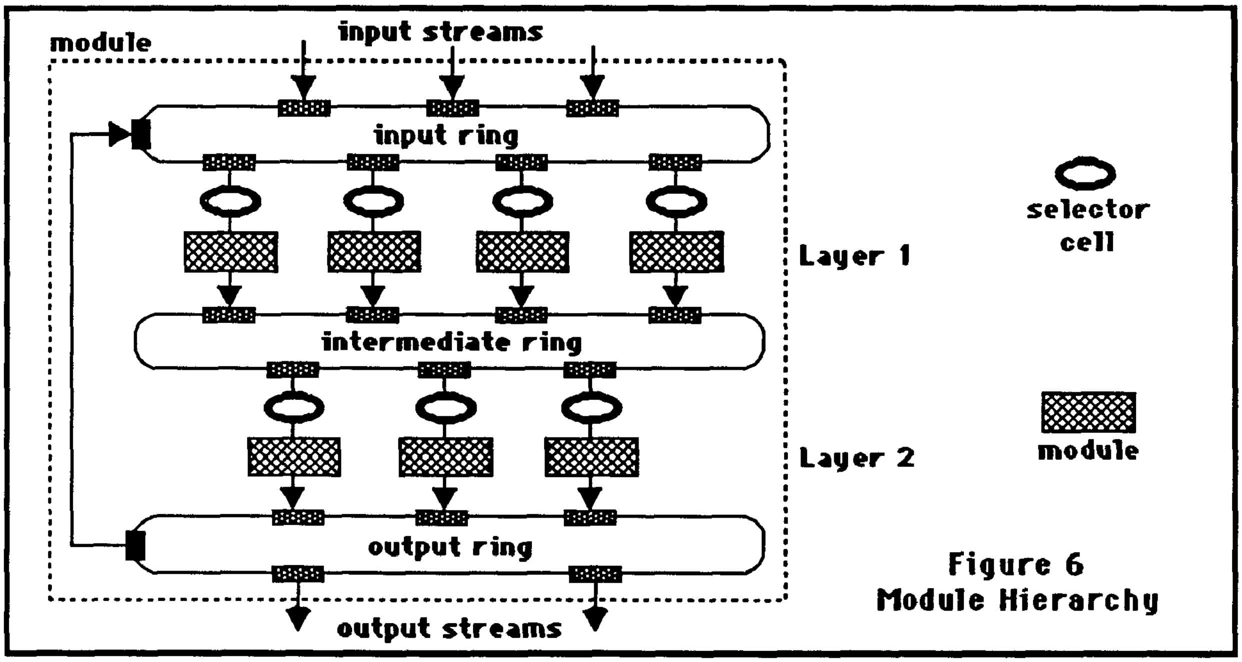


Figure 6
Module Hierarchy

The hierarchical structure of a network of modules for a collective learning system may be expressed as a context-free language generated by a formal grammar in Backus-Naur Form, as specified in Figure 7. The grammar is inherently recursive, yielding a robust and highly flexible hierarchical network architecture.

```
<network> ::= <module> <network> | <module>
<module> ::= <input ring> [<parallel cells>] <body>
             <output ring> <interface> <feedback path> <module>*
<body> ::= <parallel modules> <ring> <body> | <parallel modules>
<input ring> ::= <ring>
<output ring> ::= <ring>
<ring> ::= <interface> <link> <ring>*
<parallel modules> ::= <module> <parallel modules> | <module>
<parallel cells> ::= <cell> <parallel cells> | <cell>
```

*Circular references imply indefinite recirculation on the rings and via feedback.

Figure 7 - Grammar for Collective Learning Systems Networks

In summary, a collective learning system may be configured as a network of learning and non-learning cells assembled into modules which are hierarchically layered in such a way as to enable them to exchange messages and distribute the tasks of the system in an appropriate manner. In this way the system as a whole can organize itself and learn to give complex and accurate responses to input stimuli. All knowledge is captured in the connections among the cells (connectivity), non-learning cell functions (algorithms), the structures of the individual learning cell memories (stimulus domains and response ranges), and the evolving contents of these memories (learned strategies).

Project ALIAS, described in the next section of this paper, requires a single primitive module consisting of 32 non-learning cells and 32 learning cells arranged in two layers. It has been estimated that the human cerebral cortex contains the functional equivalent of as many as 100,000,000 cells [BOCK188, CRAS86, BRAIN79, MOUNT78]. Thus the single module used by Project ALIAS comprises about one ten-millionth of the capacity of the cerebral cortex, equivalent in processing capability to about 3000 neurons, and spatially equivalent to a cube of brain tissue about 0.3 millimeters on a side.

PROJECT ALIAS

A difficult vision task, not handled well by classical image processing algorithms, is the detection of anomalous features in images: "What's wrong with this picture?". Using a non-algorithmic approach, Project ALIAS (Adaptive Learning Image Analysis System) is applying Collective Learning Systems Theory to this task using a two-layer hierarchical network of 32 learning and 32 non-learning cells. Currently ALIAS is limited to the translation and scale invariant detection of the location and shape of anomalies. Future enhancements will include rotational invariance, as well as the automatic classification of images.

In an effort to imitate spatial geometries which occur in nature, one set of test images presented to ALIAS consists of 2-dimensional samples of 3-dimensional fractal objects. These objects are algorithmically derived from uniform power distributions in the frequency domain, subjected to inverse-frequency filtering, and moved to the spatial domain by the application of the inverse discrete Fourier transform. The spatial complexity of these 2-dimensional samples (e.g., slices) can be theoretically predicted and verified, assuring rigorous parametric control of the feature detection capabilities of ALIAS.

From each high-resolution image ALIAS derives a set of transformed images at several *ranks* presenting various spatial perspectives of the original image. During unsupervised learning ALIAS acquires a Normality Hypothesis for each transformed image (rank). This Normality Hypothesis allows ALIAS to estimate the location and shape of anomalies purposely introduced into otherwise "normal" images. During supervised learning the accuracy of each collection of estimates is reported by the evaluating Environment to the learning cells, which use these evaluations to refine their Normality Hypotheses accordingly.

Non-Learning Processes

Each image presented to ALIAS is a 252 by 252 matrix of binary pixels (black or white). The first layer of ALIAS is the Transformation Layer. It consists of 32 non-learning cells in parallel. These cells transform the original image into 32 different "spatial perspectives", called ranks. Because the existing ALIAS hardware system consists of exactly 32 Transputers, ALIAS is currently limited to processing 32 different ranks simultaneously. Additional Transputers would allow more ranks to be processed in parallel, but initial results indicate that additional ranks would probably not provide a significant improvement in performance.

Currently, the Transformation Layer employs a very simple transformation, called Tile Reduction. A transformed image under Tile Reduction results from reducing contiguous, non-overlapped regions of the original image to one bit of information. Hereafter, the Transformation Layer will be referred to as the Reduction Layer. Tile Reduction produces 32 ranks of decreasing spatial resolution by computing the ratio of white to black pixels within the Reduction Token, a square matrix of pixels whose size is equal to the rank number. If the number of white pixels within the Reduction Token exceeds the number of black pixels, the corresponding block in the reduced image is set to white; otherwise it is set to black, as shown in Figure 8.

Using Tile Reduction, the size of the Reduction Token may be varied from 1 to 63. At rank 1 the Reduction Token is a 1 by 1 matrix, and thus no reduction is required; each block is identical in size and color (black or white) to the corresponding pixel in the original image. As the rank increases by one, the width and height of the Reduction Token increase by one. At rank 2 the Reduction Token is a 2 by 2 matrix of pixels, so that the reduced block is white if at least 3 pixels in the Reduction Token are white, and black otherwise. At rank 36 the Reduction Token is a 36 by 36 matrix; the reduced block is white if at least 648 pixels in the Reduction Token are white, and black otherwise.

The Reduction Token is scanned across the entire image from left to right and from top to bottom in a non-overlapped manner. The reduced image is a matrix of blocks which are used to assemble the Analysis Tokens used by the Analysis Layer to generate the Normality Hypothesis. An example of an original fractal-based image (rank 1) and its reduced image at rank 10 is shown Figure 9.

During supervised learning when an anomaly is introduced into a normal image, it is reduced along with the rest of the pixels in the image at all 32 ranks. If it is small, it will disappear after a few ranks; if it is large enough, it will still be present in the reduced image at rank 32.

Tile Reduction does not preserve spatial phase across all ranks, and initial results indicate that this distortion degrades the performance of ALIAS. Therefore, the research team is investigating alternative transformations which employ heavily overlapped scans and/or comprehensive sampling schemes, and which will produce a reasonably consistent number of Analysis Tokens at all ranks, which offers significant statistical sampling advantages.

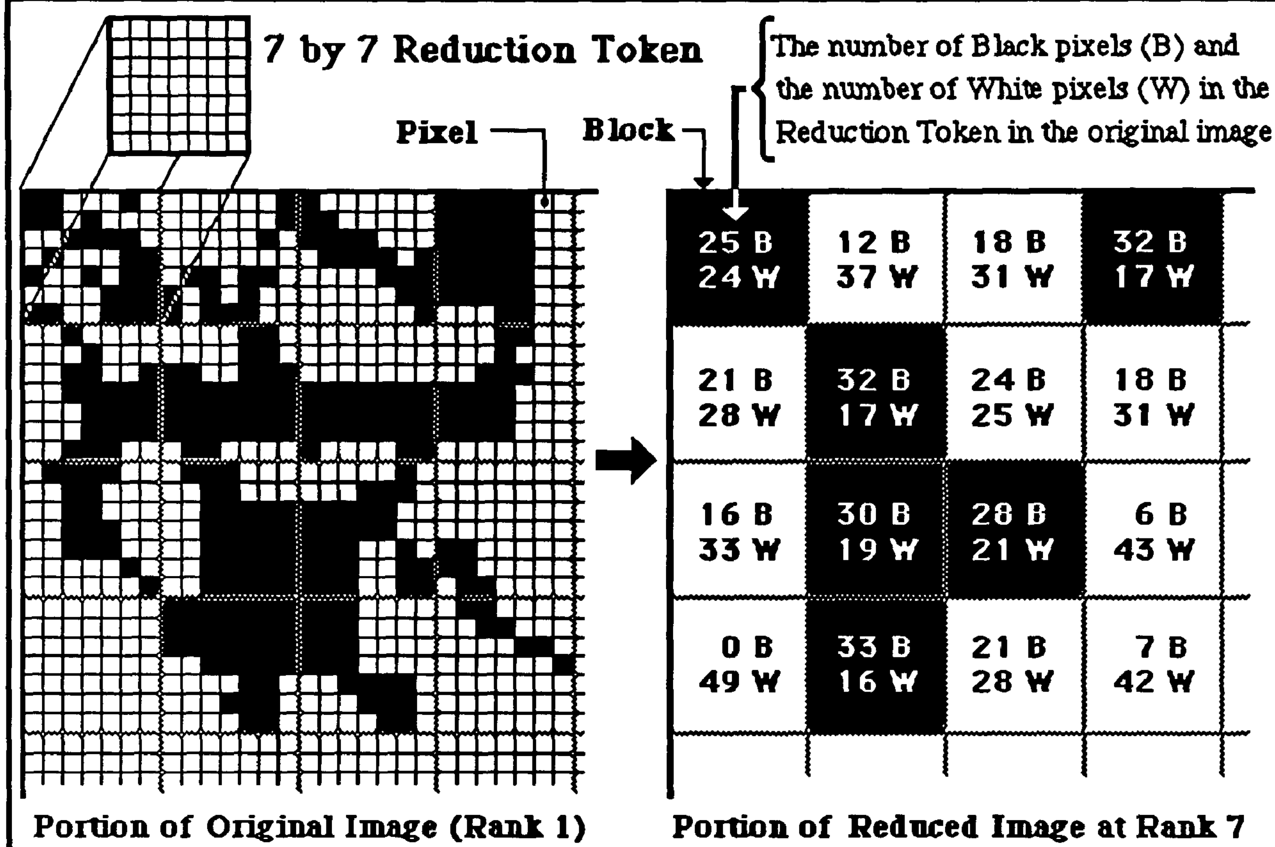


Figure 8 - TILE Image Reduction Technique

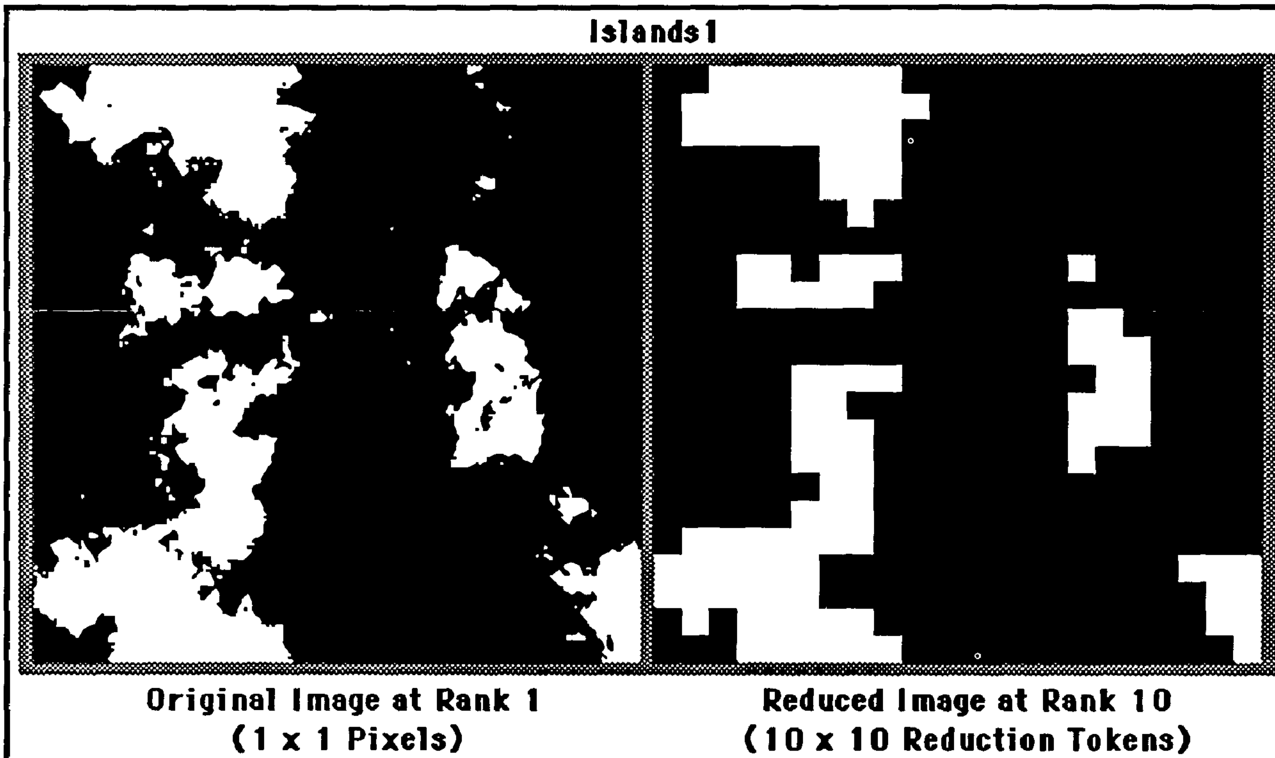


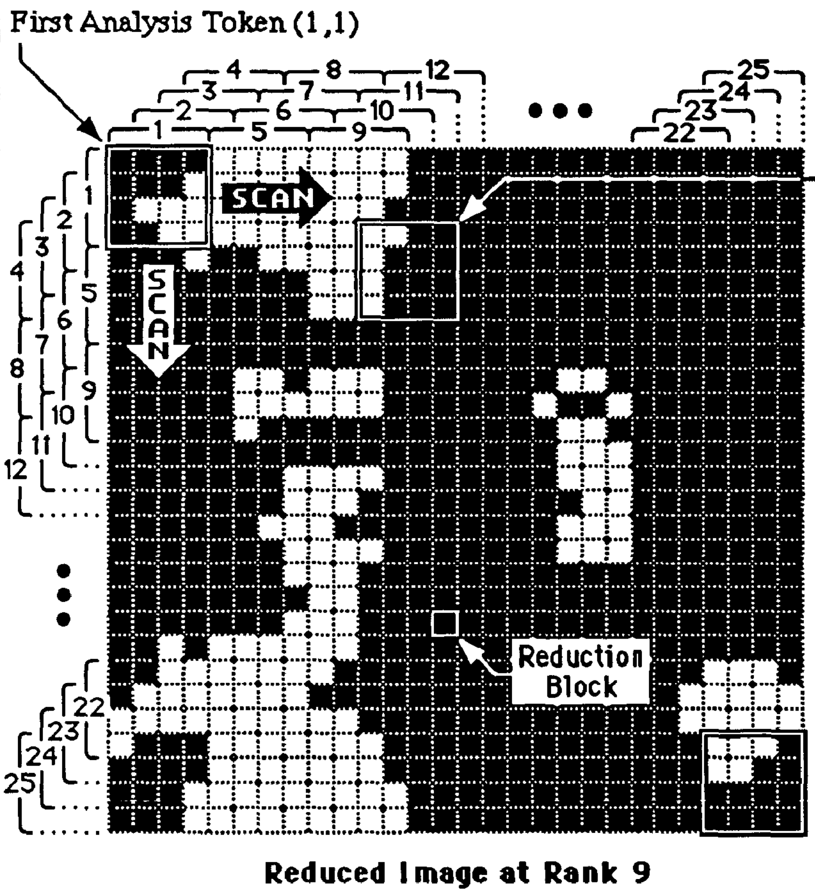
Figure 9 - Example of an Original and Reduced Image (TILE Reduction)

After a cell in the Reduction Layer has transformed the original image to a matrix of reduced blocks, ALIAS uses this reduced image to produce a stream of Analysis Tokens for the corresponding rank. The Normality Hypothesis for this rank is based upon the Analysis Patterns captured by each of these Analysis Tokens. An Analysis Token is a 4 by 4 matrix of reduced blocks. Beginning at the upper left-hand corner of the reduced image, ALIAS shifts the Analysis Token to the left, one block at a time. At the end of a row of blocks, ALIAS returns to the left edge of the reduced image, shifts the Analysis Token down one block, and scans the next row of blocks. This scanning process is repeated until the last Analysis Token has captured the Analysis Pattern in the lower right-hand corner of the reduced image. This process is illustrated in Figure 10.

This heavily overlapped scanning methodology guarantees that each block of the reduced image will be captured in 16 different spatial contexts. This also causes the number of Analysis Tokens produced to vary with rank: 62001 at rank 1, 15129 at rank 2, 6561 at rank 3, 3600 at rank 4, etc. Rank 63 produces exactly one Analysis Token. For speed, ALIAS transforms the image and produces the Analysis Tokens for all 32 different ranks in parallel.

Each Analysis Pattern is 16-bit vector which is a row-major encoding of the 4 by 4 matrix of the reduction blocks comprising each Analysis Token. Thus there are 65536 possible Analysis Patterns (2 to the power 16). The Analysis Pattern captured by each Analysis Token produced at the Reduction Layer becomes a stimulus for a learning cell at the corresponding rank in the Analysis Layer to generate its Normality Hypothesis.

Figure 10
Analysis Token Production



Reduced Image at Rank 9

Sample Analysis Patterns

col,row	Pattern
1, 1	1111 1110 1000 1100
2, 1	1110 1100 0000 1000
3, 1	1100 1000 0000 0000
4, 1	1000 0000 0000 0000
1, 2	1110 1000 1100 1110
2, 2	1100 0000 1000 1101
3, 2	1000 0000 0000 1011
4, 2	0000 0000 0000 0110
...	...
11, 4	0011 0111 0111 0111
...	...
25, 25	0001 0011 1111 1111

Learning Processes

The output from ALIAS for each image at each rank is the Normality Hypothesis. This hypothesis is a vector of learned estimates of the degree of normality for each Analysis Pattern encountered in a set of transformed images at a particular rank, i.e., whether or not the Analysis Pattern "belongs" in images of the particular class at the particular rank. ALIAS displays the Normality Hypothesis for each rank as an "image" in which the gray level represents the value of the Normality for each Analysis Pattern which occurred, with white representing completely normal, black representing completely abnormal, and grays lying somewhere in between.

The weighted average of all of the Normality estimates for all 32 ranks for each location in the image is also displayed. This representation is known as the Super Hypothesis. Both the Rank Hypothesis and Super Hypothesis are used by the Environment to compute the collective evaluation for the learning cells. This evaluation is fed back to the cells at regular intervals (the collection length) and is used by the cells to measure the error of their individual hypotheses and update their Normality estimates accordingly.

Unsupervised Learning

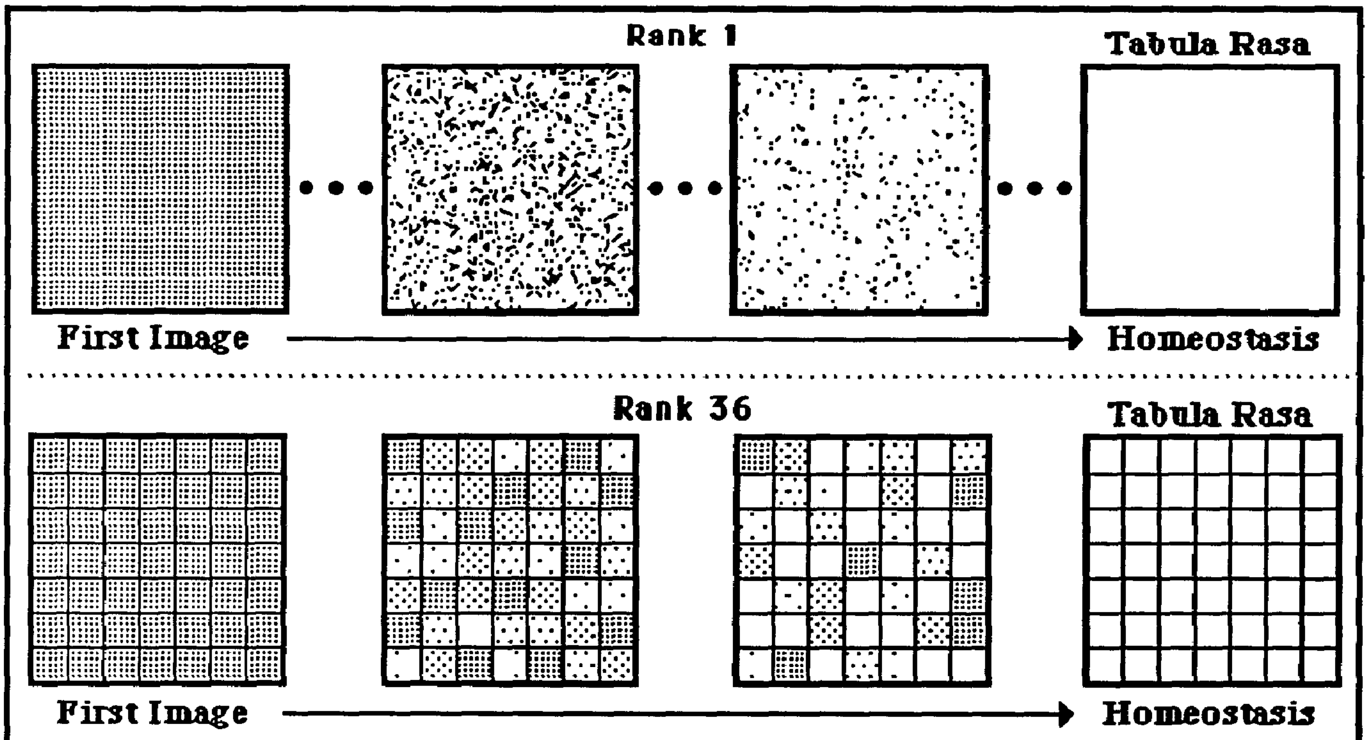
During unsupervised learning ALIAS is presented with images which do not contain anomalies - - - textbook learning. The memories of the learning cells steadily accumulate a statistical basis sufficient for characterizing the syntactic structure of the features comprising a particular class of normal images at all ranks (homeostasis). Obviously care must be taken to assure that all images presented during unsupervised learning are free of intentional anomalies and belong to the same syntactic class, e.g., are different samples derived in the same way from the same fractal object with the same fractal dimension. ALIAS can, however, learn to ignore random inconsistencies in the normal images, exhibiting excellent immunity to noise.

Whenever a specific feature appears in a normal image, the learning cells increase the probability that the feature is, indeed, normal for the particular class of images; this modification process is called Discovery. Whenever a feature which has been previously "discovered" fails to appear in an image, the learning cells decrease the probability that the feature is normal for the image class; this modification process is called Extinction.

As the memories of the learning cells converge to homeostasis during unsupervised learning, the displays of the Normality Hypotheses will mirror this process by converging to blank (all-white) slates (Tabula Rasa) as shown in Figure 11. When homeostasis for the image class has been achieved, the contents of these memories may then be used as a baseline for supervised learning, during which ALIAS learns to determine the location and shape of various anomalous features which have been intentionally inserted into normal images of the particular class.

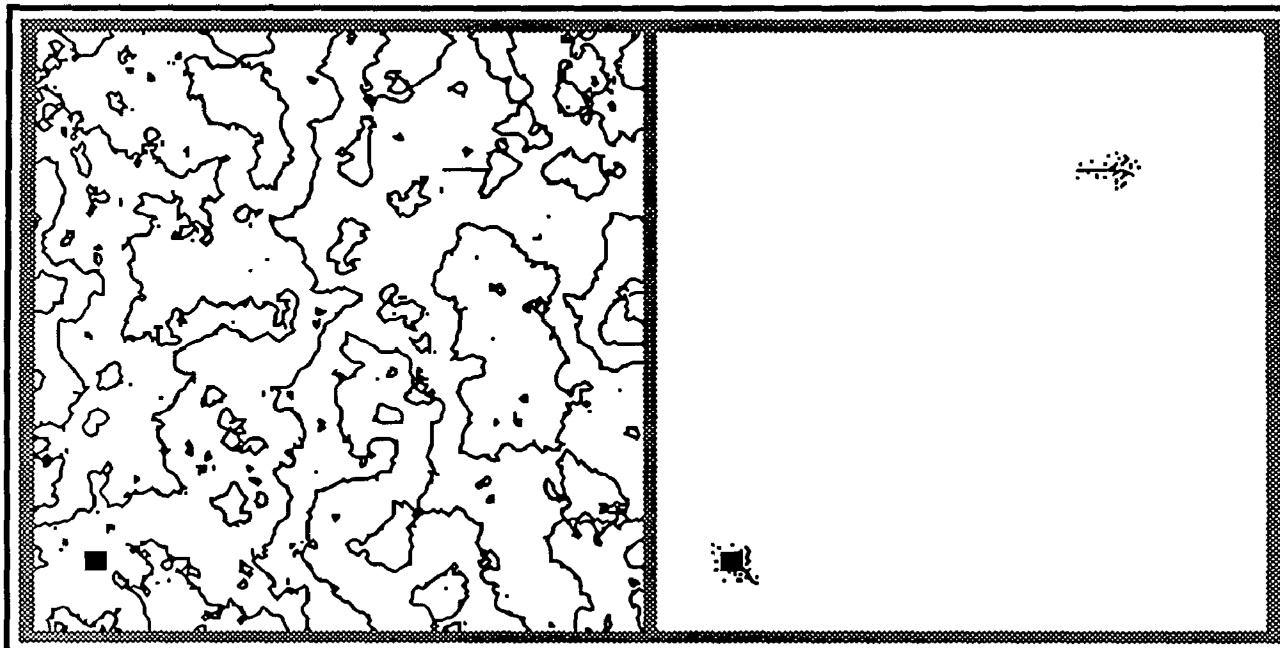
Supervised Learning

During supervised learning one or more anomalous features are introduced into instances of otherwise normal images, and these "corrupted" images are presented to ALIAS. As before, the Analysis Layer generates a Normality Hypothesis for each image at each rank and submits it to the Environment for display and evaluation, as shown in Figure 12. Unlike unsupervised learning, however, the Environment will return *negative* evaluations for those anomalies which have been specified as *abnormal*, as well as positive evaluations for those anomalies which have been specified as normal. Thus, during supervised learning ALIAS can learn to classify unusual features (anomalies for the image class) as either normal or abnormal, instead of being limited to learning them as simply more-or-less normal, as it is by default during unsupervised learning.



(Normality Tokens are actually displayed as gray levels)

Figure 11 - Evolution of the Tabula Rasa during Unsupervised Learning



Original Image at Rank 1
Anomalous 'Icepack5'

Normality Hypothesis at Rank 1
Location and Shape of Anomalies

Figure 12 - Sample Normality Hypothesis for an Anomalous Image

When the ALIAS Environment creates a test anomaly and introduces it into an image, the location, exact geometry, and normality or abnormality of every spatial element is known. Thus the Environment can issue highly precise and accurate evaluations of the resulting Normality Hypotheses. Note, however, that humans acting as the Environment may not know the normality of all the spatial elements of anomalies which they introduce, and thus may not be able to give accurate evaluations. Nevertheless, based on the evaluation received at the end of a stage, the learning cells apply their compensation functions and update the appropriate elements of the Normality at each rank. The modification of learning cell memories based on external evaluations is called Conditioning. If the evaluations are reasonably accurate, by means of Conditioning ALIAS can learn to accept or reject a wide variety of anomalous features of various shapes and sizes which are not typically present in the normal image set presented during unsupervised learning.

Learning Cell Memory

The Analysis Layer of ALIAS consists of 32 learning cells in parallel, one for each of 32 ranks chosen from all possible ranks, as determined by the Transformation scheme (Tile Reduction offers 63 possible ranks). Each cell generates a Normality Hypothesis from the current Normalities stored in its memory, i.e., its State Transition Matrix (STM). The stimulus domain of the STM for each learning cell at each rank consists of 65536 stimulants (Analysis Patterns). Its organization is shown in Figure 13.

When a learning cell in the Analysis Layer receives a stimulus (Analysis Pattern), it immediately transmits the corresponding value of the Normality in its STM to the Environment. In addition, the learning cell increments the corresponding value of the Pattern-Occurrence Count in its STM by one. The Pattern-Occurrence Count may be used by the cell in its compensation policy as a measure of the extent of its "experience" with the particular Analysis Pattern to temper its reaction to incoming evaluations. For example, the more experience a cell has had with a particular Analysis Pattern, the more seriously it might take its evaluations, implying a larger change (positive or negative) in the corresponding Normality. Experience with Collective Learning shows conclusively that a judiciously chosen modification policy can accelerate learning by several orders of magnitude. [e.g., PLOURDE78, WHITE89]

During supervised learning, the processes of Discovery and Extinction are still active. In addition, the Normality is increased (Positive Conditioning) or decreased (Negative Conditioning) based on periodic (collective) evaluations received from the Environment. Like Discovery, Positive Conditioning drives the Normality toward +32767 as a limit. Positive Conditioning may be used to "condition" ALIAS to recognize certain anomalies as normal, such as airports in the middle of deserts, eyeglasses on faces, and ships on oceans. Unlike Extinction, however, Negative Conditioning drives the Normality toward -32767 as a limit. Negative Conditioning is the only mechanism by which features can be classified as abnormal; in the absence of negative evaluations from the Environment, features can only be classified between "unknown" (Normality = 0) and "completely normal" (Normality = +32767).

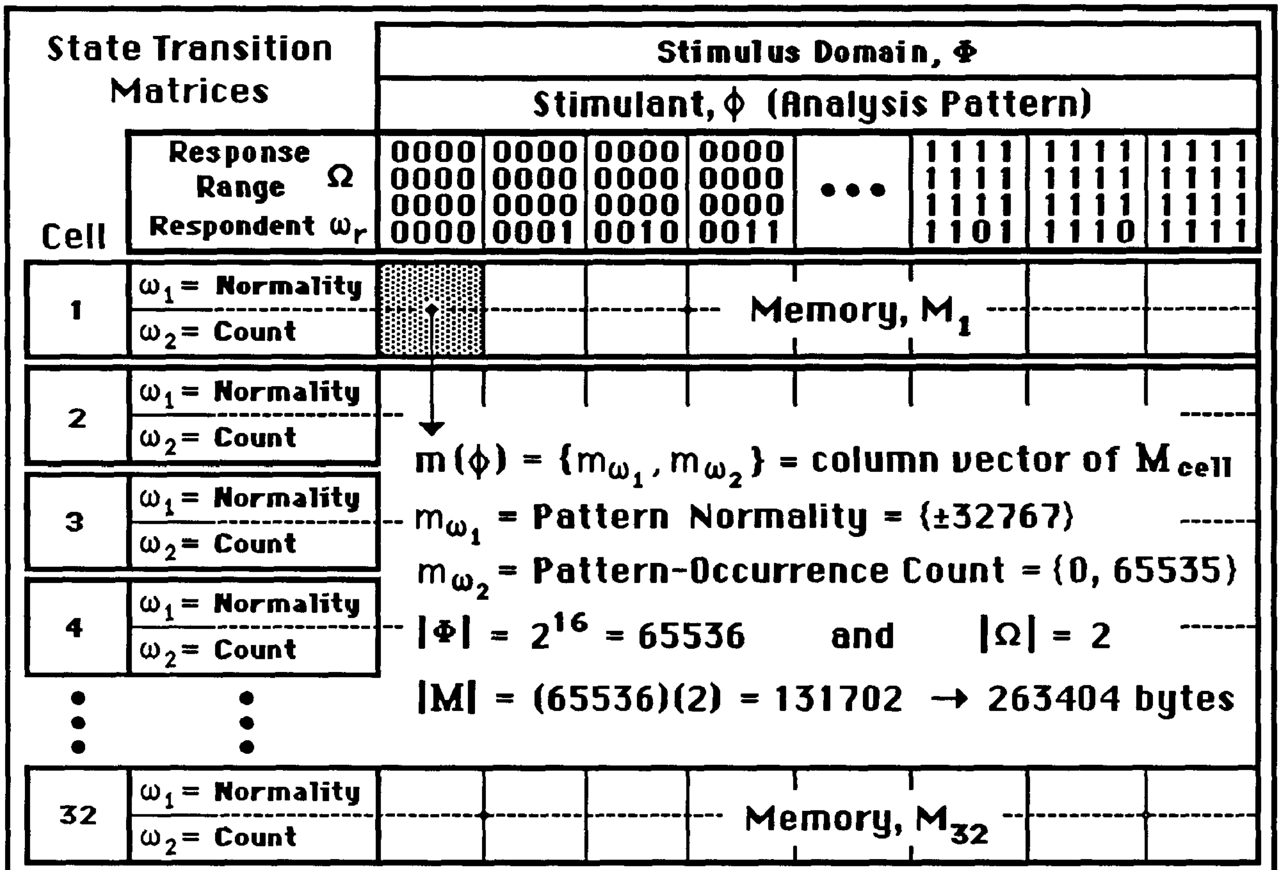


Figure 13 - Learning Cell Memory Organization

Network Implementation

Logical Topology

ALIAS requires a single module consisting of 32 non-learning cells and 32 learning cells arranged in two layers: the Reduction Layer and the Analysis Layer. The logical topology is shown in Figure 14. The Input Ring distributes the original image to the non-learning cells in the Reduction Layer in packets containing the stream of pixels necessary to produce one complete row of Reduction Tokens (swath). After reducing each swath, the cell uses the resulting Reduction Tokens as components (blocks) for the production of the Analysis Tokens, placing each Analysis Token on the Intermediate Ring as it is produced.

The Intermediate Ring distributes the Analysis Tokens for each rank to the corresponding learning cell in the Analysis Layer. As each Analysis Token (stimulus) is received, the cell accesses its STM (memory), increments the Count for the corresponding Analysis Pattern (stimulant), and inserts the corresponding Normality Hypothesis Token on the Output Ring, which forwards it to the Environment for display and evaluation at the end of the stage. At the end of a stage, the Environment issues this evaluation to the learning cells via the Input Ring.

Physical Network Architecture

The heart of the ALIAS configuration is a network of 32 Transputers functioning as a single module. Manufactured by INMOS, Transputers are high-speed RIS processors designed to be embedded in parallel processing networks. Each Transputer has four ports with which it can communicate with other Transputers or external devices.

In order to minimize the communication and processing delays in the parallel network architecture, the physical topology of the network is somewhat different from its logical topology. The connectivity of the ALIAS Transputer network shown in Figure 15 was determined on the basis of an *a priori* analysis of the anticipated communication and processing requirements. In a more ambitious future experiment, the optimal network configuration, as well as a more efficient allocation of the cell memories, could also be learned by adaptive selector cells.

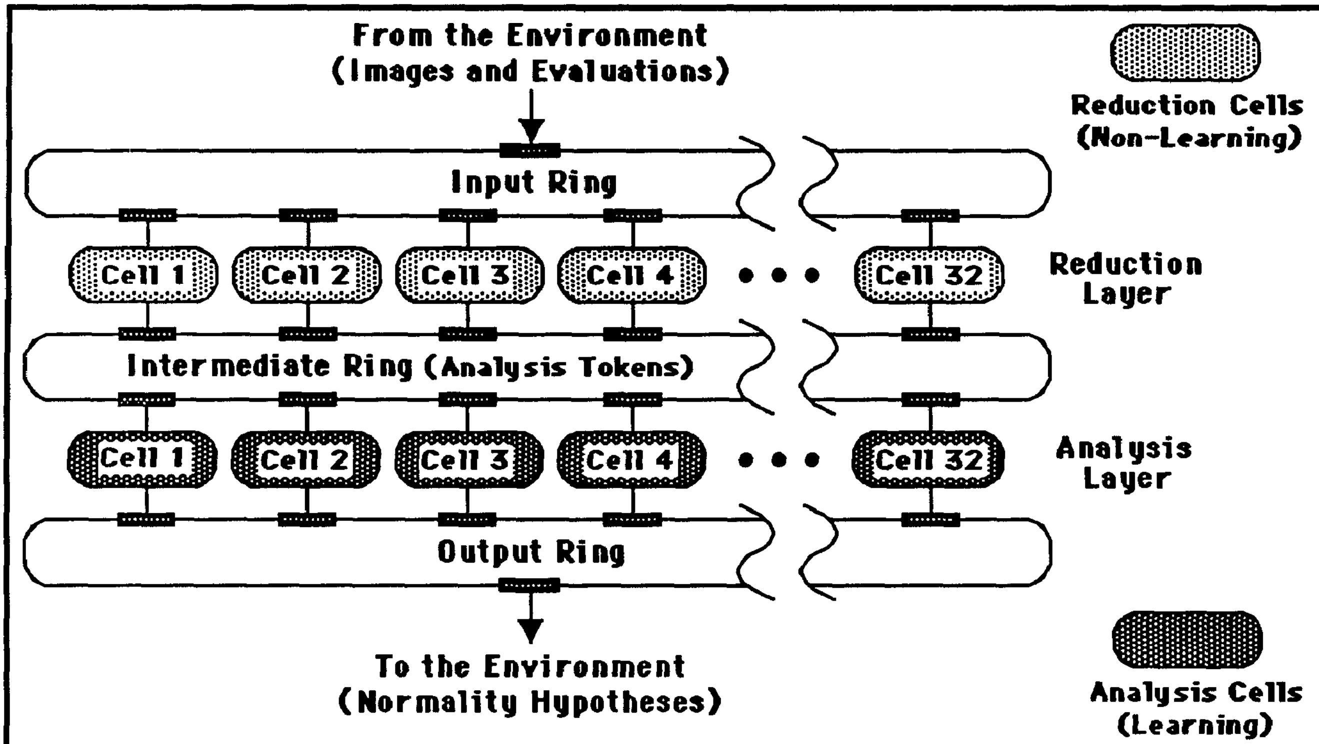


Figure 14 - Network Logical Topology

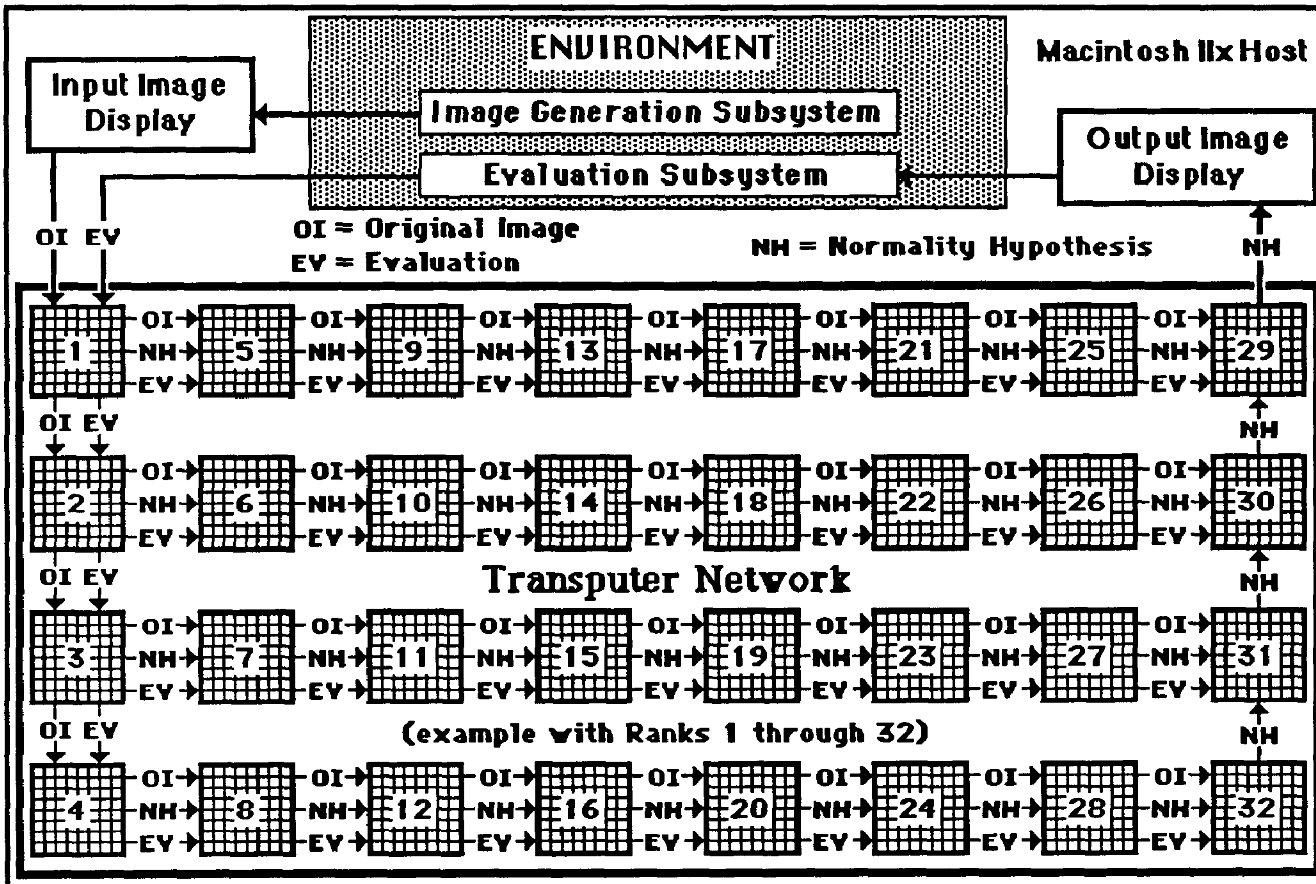


Figure 15 - Physical Network Architecture

The Environment for ALIAS resides in a Macintosh IIx (the Host), which generates input images, displays the Normality Hypotheses, evaluates these hypotheses, and monitors the operation of the system. The Image Generation Subsystem either automatically generates test images or accepts manually generated images, with or without anomalies. Each original input image is pipelined to every Transputer in the network. For its particular rank, each Transputer transforms the image, produces the analysis tokens, synthesizes the Normality Hypothesis, and submits it to the Host for display and evaluation. At the end of each stage the Evaluation Subsystem in the Host submits collective evaluations to all the Transputers, which use these evaluations to update their memories (STMs).

A friendly and powerful, icon-based user interface allows the experimenter to define complex experiment scripts for unattended operation to automatically submit specific sequences of images to ALIAS, vary the learning parameters, introduce anomalies into selected images at predetermined times, and record intermediate and final results for subsequent statistical analysis. The experimenter may also observe system performance and the progress of learning in real time on several color graphic displays.

CONCLUSIONS

During unsupervised learning ALIAS is exposed to a large number of instances of a class of normal high-resolution binary images. In the case of fractal-based test images, the complexity and syntax of the images are known and consistent. Eventually ALIAS reaches homeostasis, displaying a blank image (Tabula Rasa), signifying its recognition of normality for this class. Subsequently, during supervised learning ALIAS is exposed to instances of the Tabula Rasa containing one or more anomalous features of various shapes, sizes, and distributions, some intentionally normal, others intentionally abnormal. For each such test image at all ranks ALIAS displays the hypothesized locations of those features which it considers anomalous, as well as the associated severities. On the basis of the accuracy of the Super Hypothesis, the Environment periodically issues an evaluation to the learning cells, which they use to modify their understanding of normality. Repetitions of this process will allow ALIAS to characterize the normality of a wide variety of features and image classes: "This test image appears to be a corruption of a normal desert scene with the locations and severities of the hypothesized abnormalities indicated on the output image."

Using the Tile Reduction scheme, the analysis of a single test image is accomplished in about one second. A stable estimate of the normal salient features of a Tabula Rasa (homeostasis) is achieved in a few minutes. These times are comparable with human requirements for the same task. Initial results with the Tile Reduction scheme demonstrate excellent detection and localization, and acceptable discrimination. A series of very recent experiments indicate that more complex transformation schemes and non-linear progressive compensation policies result in vastly improved discrimination (elimination of false alerts). The ALIAS research team expects to publish a summary of useful transformation schemes and the corresponding experimental results in the next few months.

With the advent of new hardware technologies, it may soon become possible to implement extremely large adaptive networks to simulate very high level cognitive functions as exhibited by the human cerebral cortex. Such technologies are typically just around the corner, their development driven by the needs of the designer. What seemed like an impossibility before the advent of solid-state technology and micro-electronics, now becomes a glimmer of hope on the technological horizon.

BIBLIOGRAPHY

[BOCK76] P. Bock, "Observation of the Properties of a Collective Learning Stochastic Automaton", Proc. International Information Sciences Symp, Patras, Greece, Aug 76.

[BOCK188] "A Perspective on Artificial Intelligence: Learning to Learn", Annals of Operations Research, v16, Apr 88.

[BOCK288] "A Massively Parallel Architecture for Collective Learning Systems", Journal of the Latin American Society on Artificial Intelligence, Mar 88.

[BRAIN79] Hubel, et. al., The Brain, Scientific American Book, W.H. Freeman, 1979.

[BWW85] P. Bock, F. Weingard, and J. White, "A Universal Model for the Structure and Function of Collective Learning Systems", Proc. International Symp New Directions in Computing, Trondheim, Norway, Aug 85.

[CRAS86] F. Crick and C. Asanuma, "Certain Aspects of the Anatomy and Physiology of the Cerebral Cortex", from chapter 20 of Parallel Distributed Processing (Vol 2) (D.E. Rumelhardt and J.L. McClelland, editors), The MIT Press, 1986.

[LAK81] S. Lakshmivarahan, Learning Algorithms Theory and Applications, Springer-Verlag, 1981.

[MOUNT78] V. B. Mountcastle, "An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System", from The Mindful Brain, The MIT Press, 1978.

[NAR77] K. S. Narendra, Editor, "Special Issue on Learning Automata", Jour of Cybernetics and Info Sci, v1 #2-4, 1977.

[PLOURDE78] R. Plourde, "A Progressive Compensation Policy for Collective Learning Automata", Masters Thesis under Professor Peter Bock for the Department of Electrical Engineering and Computer Science at the George Washington University, Washington DC, 1978.

[WHITE89] J. White, "A Modification Policy that Accelerates the Convergence of Collective Learning Systems' Optimality", Doctoral Dissertation under Professor Peter Bock for the Department of Electrical Engineering and Computer Science at the George Washington University, Washington DC, 1989.