Communicated by Fernando Pineda

Working Memory Networks for Learning Temporal Order with Application to Three-Dimensional Visual Object Recognition

Gary Bradski
Gail A. Carpenter
Stephen Grossberg
Center for Adaptive Systems and Department of Cognitive and Neural Systems, Boston University, Boston, MA 02215 USA

Working memory neural networks, called Sustained Temporal Order REcurrent (STORE) models, encode the invariant temporal order of sequential events in short-term memory (STM). Inputs to the networks may be presented with widely differing growth rates, amplitudes, durations, and interstimulus intervals without altering the stored STM representation. The STORE temporal order code is designed to enable groupings of the stored events to be stably learned and remembered in real time, even as new events perturb the system. Such invariance and stability properties are needed in neural architectures which self-organize learned codes for variable-rate speech perception, sensorimotor planning, or three-dimensional (3-D) visual object recognition. Using such a working memory, a self-organizing architecture for invariant 3-D visual object recognition is described. The new model is based on the model of Seibert and Waxman (1990a), which builds a 3-D representation of an object from a temporally ordered sequence of its two-dimensional (2-D) aspect graphs. The new model, called an ARTSTORE model, consists of the following cascade of processing modules: Invariant Preprocessor → ART 2 → STORE Model → ART 2 → Outstar Network.

1 Introduction

Working memory is the type of memory whereby a telephone number, or other novel temporally ordered sequence of events, can be temporarily stored and then performed (Baddeley 1986). Working memory, a kind of short-term memory (STM), can be quickly erased by a distracting event, unlike long-term memory (LTM). There is a large experimental literature about working memory, as well as a variety of models (Atkinson and Shiffrin 1971; Cohen and Grossberg 1987; Cohen *et al.* 1987; Elman 1990; Grossberg 1970, 1978a,b; Grossberg and Pepe 1971; Grossberg and Stone

1986; Gutfreund and Mezard 1988; Guyon *et al.* 1988; Jordan 1986; Reeves and Sperling 1986; Schreter and Pfeifer 1989; Seibert 1991; Seibert and Waxman 1990a,b; Wang and Arbib 1990).

The present class of models, called STORE (Sustained Temporal Order REcurrent) models, exhibit properties that have heretofore not been available in a dynamically defined working memory. In particular, STORE working memories are designed to encode the invariant temporal order of sequential events, or items, that may be presented with widely differing growth rates, amplitudes, durations, and interstimulus intervals. The STORE model is also designed to enable all possible groupings of the events stored in STM to be stably learned and remembered in LTM, even as new events perturb the system. In other words, these working memories enable chunks (also called compressed, categorical, or unitized representations) of a stored list to be encoded in LTM in a manner that is not erased by the continuous barrage of new inputs to the working memory.

Working memories with these properties are important in many applications wherein properties of behavioral self-organization are needed. Three important applications are real-time self-organization of codes for variable-rate speech perception, sensorimotor planning, and 3-D visual object recognition. Architectures for the first two types of application are described in Cohen *et al.* (1987) and Grossberg and Kuperstein (1989). Herein we outline how such a working memory can both simplify and extend the capabilities of the Seibert and Waxman model for 3-D visual object recognition (Seibert and Waxman 1990a,b; Seibert 1991).

2 Invariance Principle and Partial Normalization

The STORE neural network working memories are based on algebraically characterized working memories that were introduced by Grossberg (1978a,b). These algebraic working memories were designed to explain a variety of challenging psychological data concerning working memory storage and recall. In these models, individual events are stored in working memory in such a way that the pattern of STM activity across event representations encodes both the events that have occurred and the temporal order in which they have occurred. In the cognitive literature, such a working memory is often said to store both *item* information and *order* information (Healy 1975; Lee and Estes 1981; Ratcliff 1978). The models also include a mechanism for reading out events in the stored temporal order. An event sequence can hereby be performed from STM even if it is not yet incorporated through learning into LTM, much as a new telephone number can be repeated the first time that it is heard.

The large data base on working memory shows that storage and performance of temporal order information from working memory are not always veridical (Atkinson and Shiffrin 1971; Baddeley 1986; Reeves and

Sperling 1986) These deviations from veridical temporal order in STM could be explained by the algebraic working memory model as consequences of two design principles that have clear adaptive value. These principles are called the Invariance Principle and Partial Normalization (Grossberg 1978b).

2.1 Invariance Principle. The spatial patterns of STM activation across the event representations of a working memory are stored and reset in response to sequentially presented events in such a way as to leave the temporal order codes of all past event groupings invariant.

In particular, a temporal list of events is encoded in STM in a way that preserves the stability of previously learned LTM codes for familiar sublists of the list. For example, suppose that the word MY has previously been stored in a working memory's STM and has established a learned chunk in LTM. Suppose that the word MYSELF is then stored for the first time in STM. The word MY is a syllable of MYSELF. The STM encoding of MY as a syllable of MYSELF may not be the same as its STM encoding as a word in its own right. On the other hand, MY's STM encoding as part of MYSELF should not be allowed to force forgetting of the LTM code for MY as a word in its own right. If it did, familiar words, such as MY, could not be learned as parts of larger words, such as MYSELF, without eliminating the smaller words from the lexicon. More generally, new wholes could not be built from familiar parts without erasing LTM of the parts.

The Invariance Principle can be algebraically realized as follows, provided that no list items are repeated. Assume for simplicity that the ith list item is preprocessed by a winner-take-all network. Each list item then activates a single output node of the preprocessor network. Properties of the working memory also hold if a finite set of output nodes is activated for each item. The winner-take-all case is described herein for notational simplicity. Let the winner-take-all node that is activated by the ith item send a binary input I_i to the first working memory level F_1 (Fig. 1). Let x_i denote the activity of the ith item representation of F_1. Suppose that I_i is registered in working memory at time t_i. At time t_i, the activity pattern $[x_1(t_i), x_2(t_i), \ldots, x_n(t_i)]$ across F_1 stores the effects of the list $I_1, I_2, \ldots, I_i$ of previous inputs. The input I_i updates the activity values $x_k(t_{i-1})$ to new values $x_k(t_i)$ for all nodes $k = 1, 2, \ldots, i$ according to the following rule:

$$x_k(t_i) = \begin{cases} 0 & \text{if } k > i \\ \mu_i & \text{if } k = i \\ \omega_i x_k(t_{i-1}) & \text{if } k < i \end{cases} \tag{2.1}$$

At time t_i, the pattern $[x_1(t_{i-1}), x_2(t_{i-1}), \ldots, x_{i-1}(t_{i-1})]$ of previously stored STM activities is multiplied by a common factor ω_i as the ith item is instated with some activity μ_i.

The storage rule (2.1) satisfies the Invariance Principle for the following reason. Suppose that F_1 is the first level of a two-level competitive

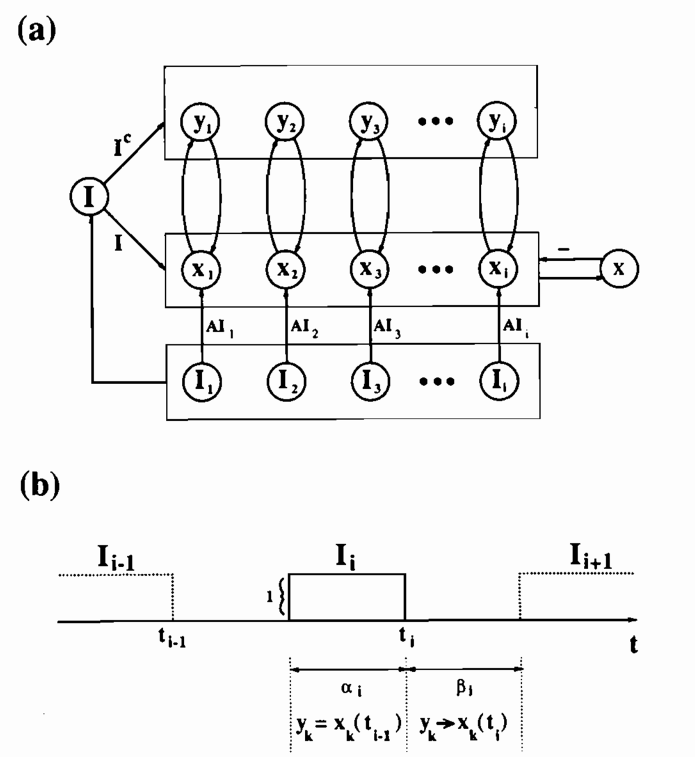

Figure 1: (a) Elementary STORE model: STM activity x_i at level 1 registers the item input I_i, nonspecific shunting inhibition x, and level 2 STM y_i. STM activity y_i at level 2 registers x_i. Complementary input-driven gain signals I and I^c control STM processing at levels 1 and 2. (b) Input $I_i(t)$ equals 1 for $t_i - \alpha_i < t \le t_i$. When all inputs are off ($t_i < t \le t_i + \beta_i$) level 2 variables y_k relax to level 1 values $x_k(t_i)$.

learning network (Grossberg 1976). Then F_1 sends signals to the second level F_2 via an adaptive filter. The total input to the jth F_2 node is $\Sigma_k x_k z_{kj}$, where z_{kj} denotes the LTM trace, or adaptive weight, in the path from the kth F_1 node to the jth F_2 node. In psychological terms, each active F_2 node represents a chunk of the F_1 activity pattern. When the jth F_2 node is active, the LTM weights z_{kj} converge toward x_k; in other words, the weight vector becomes parallel to the F_1 activity vector. When a new item is added to the list, the Invariance Principle implies that the previously active items in the list will simply be multiplied by a common factor, thereby maintaining a constant ratio between the previously active items. Constant activity ratios imply that the former F_1 activity vector remains parallel to its weight vector as its magnitude changes under new inputs. Hence, adding new list items does not invalidate the STM and LTM codes for sublists. In particular, the temporal order of items in each sublist, encoded as relative sizes of both the STM and the LTM variables, remains invariant.

2.2 Partial Normalization. The Partial Normalization rule algebraically instates the classical property of the limited capacity of STM (Atkinson and Shiffrin 1971). A convenient statement of this property is given by the equation

$$S_i \equiv \sum_k x_k(t_i) = \mu_1\theta_i + S(1-\theta_i) \tag{2.2}$$

where $\theta_1 = 1$ and θ_i decreases toward 0 as i increases. For example, let $\theta_i = \theta^{i-1}$, with $0 < \theta < 1$. Total activity S_i increases toward an asymptote, S, as new items are presented. Parameter S characterizes the "limited capacity" of STM. In human subjects, this parameter is determined by biological constraints. In an artificial neural network, parameter S can be set at any finite value.

Using equations 2.1 and 2.2, it was proved in Grossberg (1978a) that the rate at which S_i approaches its asymptote S helps determine the form of the STM activity pattern. The pattern $(x_1, \ldots, x_i)$ can exhibit primacy (all $x_{k-1} > x_k$), recency (all $x_{k-1} < x_k$), or bowing, which combines primacy for early items with recency for later items (Grossberg 1978a). These various patterns correspond to properties of STM storage by human subjects. In particular, model parameters are typically set so that the STM activity pattern exhibits a primacy gradient in response to a short list. Since more active nodes are read-out of STM before less active nodes during performance trials, primacy storage leads to the correct order of recall in response to a short list. Using the same parameters, the STM activity pattern exhibits a bow in response to longer lists, and approaches a recency gradient in response to still longer lists. An STM bow leads to performance of items near the list beginning and end before items near the list middle. A larger STM activity at a node also leads to a higher probability of recall from that node under circumstances when the network is perturbed by noise. An STM bow thus leads to earlier recall and to a higher probability of recall from items at the beginning and the end of a list. These formal network properties are also properties of data from a variety of experiments about working memory, such as free recall experiments during which human subjects are asked to recall list items after being exposed to them once in a prescribed order (Atkinson and

Shiffrin 1971; Healy 1975; Lee and Estes 1981). Effects of LTM on free recall data have also been analyzed by the theory (Grossberg 1978a,b).

The multiplicative gating in equation 2.1 and the partial normalization in equation 2.2 are algebraic versions of the types of properties that are found in a general form in shunting competitive feedback networks (Grossberg 1973). A task of the present research was to discover specialized shunting networks that realize equations 2.1 and 2.2 as emergent properties of their real-time dynamics. The STORE model is a real-time shunting network, defined below, which exhibits the desired emergent properties. In particular, the STORE system moves from primacy to bowing to recency as a single model parameter is increased.

3 Working Memories Invariant Under Variable Input Speed, Duration, and Interstimulus Interval

Two types of real-time working memories, *transient* models and *sustained* models, can realize the invariance and partial normalization properties. In a transient model, presentation of items of different durations can alter the previously stored pattern of temporal order information. Transient memory models can still accurately represent temporal order if input durations are controlled by a preprocessing stage. Sustained models allow input durations and interim intervals to be essentially arbitrary: so long as these intervals are not too short, temporal input fluctuations have no effect on patterns stored in memory. A sustained neural network model is defined below. This two-level STORE model codes lists of distinct items. A variant of the STORE model design, to be discussed in a subsequent article, can encode the temporal order of lists in which each item may occur multiple times. Each item may also be represented by multiple nodes.

The first level of the STORE model (Fig. 1a) consists of nodes with STM activity x_i. The ith item is assumed to send a unit input I_i to the ith node for a time interval of length α_i. After an interstimulus interval of length β_i, the next item sends an input to the $(i+1)$st node, and so on. Each STM node also receives shunting inhibition via a nonspecific feedback signal that is proportional to the total STM activity x. The second STORE level consists of excitatory interneurons whose activity y_i tracks x_i. A critical additional factor in the model is gain control that enables changes in x_i to occur only when an input is present and enables changes in y_i to occur only when no input is present. This alternating gain control allows feedback from y_k to x_k $(k < i)$ to preserve previously stored patterns even when a new input I_i is on for a long time interval. These processes are defined below in the simplest way possible to permit complete analysis and understanding of the model's emergent properties.

3.1 STORE model equations. The STORE model is defined by the dimensionless equations

$$\frac{dx_i}{dt} = [AI_i + y_i - x_i x]I \tag{3.1}$$

and

$$\frac{dy_i}{dt} = [x_i - y_i]I^c \tag{3.2}$$

where

$$x \equiv \sum_k x_k \tag{3.3}$$

$$I \equiv \sum_k I_k \tag{3.4}$$

$$I^c \equiv 1 - I \tag{3.5}$$

and

$$x_i(0) = y_i(0) = 0 \tag{3.6}$$

The input sequence I_i is given by

$$I_i(t) = \begin{cases} 1 & \text{if } t_i - \alpha_i < t < t_i \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

(Fig. 1b). The input durations (α_i) and the interstimulus intervals ($\beta_i = t_i - \alpha_i - t_{i-1}$) are assumed to be large relative to the dimensionless relaxation times of x_i and y_i, set equal to 1 in equations 3.1 and 3.2. Thus each x_i reaches steady state when inputs are on and each y_i reaches x_i when inputs are off. Otherwise, t_i and α_i can be arbitrary, and their values have no effect on patterns of memory storage.

3.2 Temporal order patterns. We will now examine how system properties vary as a function of the single free parameter, A, in equation 3.1. We will see that, in all cases, patterns of past activities remain invariant as new inputs perturb the system, and partial normalization obtains. In addition, the STM pattern $(x_1, \ldots, x_i)$ exhibits primacy for small A, recency for $A > 1$, and bowing for intermediate values of A, as follows.

By equations 3.1, 3.6, and 3.7, when the ith input is presented, $I_i = 1$, $y_i = 0$, and

$$x_i \to \frac{A}{x} \tag{3.8}$$

For $k < i, I_k = 0$ and

$$x_k \to \frac{y_k}{x} = \frac{x_k(t_{i-1})}{x} \tag{3.9}$$

Thus the relative sizes of the activities in pattern $(x_1, \ldots, x_{i-1})$ are preserved when x_i becomes active. Amplitudes increase uniformly if total activity $x < 1$, and decrease uniformly if $x > 1$. Equations equations 3.1 and 3.2 imply that the variable x obeys the equations

$$\frac{dx}{dt} = [A + y - x^2]I \tag{3.10}$$

and

$$\frac{dy}{dt} = [x - y]I^c \tag{3.11}$$

where

$$y \equiv \sum_k y_k \tag{3.12}$$

Since $y(0) = 0$, equation 3.10 implies that $x(t_1) = \sqrt{A}$. At time $t = t_i$, $i > 1$, equation 3.10 implies that

$$x(t_i) = \sqrt{A + y(t_i)}$$

and equation 3.11 implies that $y(t_i) = x(t_{i-1})$. Thus the total activity S_i at time t_i satisfies $S_1 \equiv x(t_1) = \sqrt{A}$ and

$$S_i \equiv x(t_i) = \sqrt{A + S_{i-1}}, \; i > 1$$

As the number of items increases, both S_i and $x(t)$ approach

$$S = .5[1 + \sqrt{1 + 4A}]$$

which is the positive solution of

$$S = \sqrt{A + S}$$

For large A, therefore, $S_1 S^{-1} \cong 1$, whereas for small A,

$$S_1 S^{-1} \cong \sqrt{A} \ll 1$$

Thus, for large A, the total STM activity is approximately normalized at all times, whereas for small A, it grows rapidly as more inputs perturb the network. Since the size of parameter A in equation 3.1 reflects the degree to which the input I_i influences the STM pattern, recency for large A (present input dominates) and primacy for small A (past activities dominate) would be intuitively predicted. In fact, for large A, the pattern of STM activity $(x_1, \ldots, x_i)$ always shows a recency gradient. For small A, the STM patterns in response to short lists show a primacy gradient. Specifically, by equations 3.8 and 3.9,

$$x_i(t_i) = \frac{A}{x(t_i)} \tag{3.13}$$

and

$$x_{i-1}(t_i) = \frac{x_{i-1}(t_{i-1})}{x(t_i)} \tag{3.14}$$

Thus at time t_i, just after I_i has been presented,

$$x_{i-1} > x_i \quad \text{iff} \quad x_{i-1}(t_{i-1}) > A \tag{3.15}$$

Thus if $x_1(t_1) > A$, $(x_1, \ldots, x_i)$ shows a primacy gradient until $x_i(t_i) \leq A$. Presenting additional inputs $I_{i+1}, I_{i+2}, \ldots$ causes the STM pattern to bow. If $x_1(t_1) \leq A$, the STM pattern always exhibits recency. Since $x_1(t_1) = \sqrt{A}$, recency occurs for all list lengths whenever $A \geq 1$, while small A values allow relatively long lists to be stored by primacy gradients. The position at which the STM pattern bows can be calculated iteratively. For example, the bow occurs at position $i = 2$ if $1 > A \geq .5(3 - \sqrt{5}) \cong 0.382$.

These properties of the STORE model are illustrated by the computer simulations summarized in Figure 2. Each row depicts STM storage of a list at a fixed value of A. In the left column, the STM vector $(x_1, x_2, \ldots, x_7)$ is depicted at times $t_1, t_2, \ldots, t_7$ when successive inputs $I_1, I_2, \ldots, I_7$ are stored. Each activity x_i is represented by the height of a vertical bar. The top row depicts a recency gradient, the seventh row a primacy gradient, and intermediate columns represent bows at each successive list position. The middle column graphs the ratios $x_i x_{i+1}^{-1}$ through time. The horizontal graphs mean that the Invariance Principle is obeyed as soon as both items in each ratio are stored. The third column graphs the growth of total activity $x(t)$ to its capacity S. The input durations α_i in equation 3.7 varied randomly between 10 and 40. Such variations in input parameters had no discernible effect on the stored STM patterns.

4 A Self-Organizing Architecture for Invariant 3-D Visual Object Recognition

The application summarized below of the STORE model illustrates how a working memory, whose analog STM weights code both order and item information, can substantially reduce the number of connections needed to solve temporal learning problems, and simplify the modeling of such processes. Seibert and Waxman (1990a,b, 1992; Seibert 1991) have developed a novel self-organizing neural network architecture for invariant 3-D visual object recognition. In response to moving objects in space, an Invariant Preprocessor in the architecture automatically generates 2-D patterns that are invariant under changes in object position, size, and orientation, and are insensitive to foreshortening effects in 3-D. These patterns form the input vectors to an ART 2 network (Carpenter and Grossberg 1987) that self-organizes learned category representations of the invariant patterns. Each category node encodes a 2-D "aspect" of the object; that is, a single category node is activated by a collection of similar 2-D views of the object. The ART 2 vigilance parameter controls how

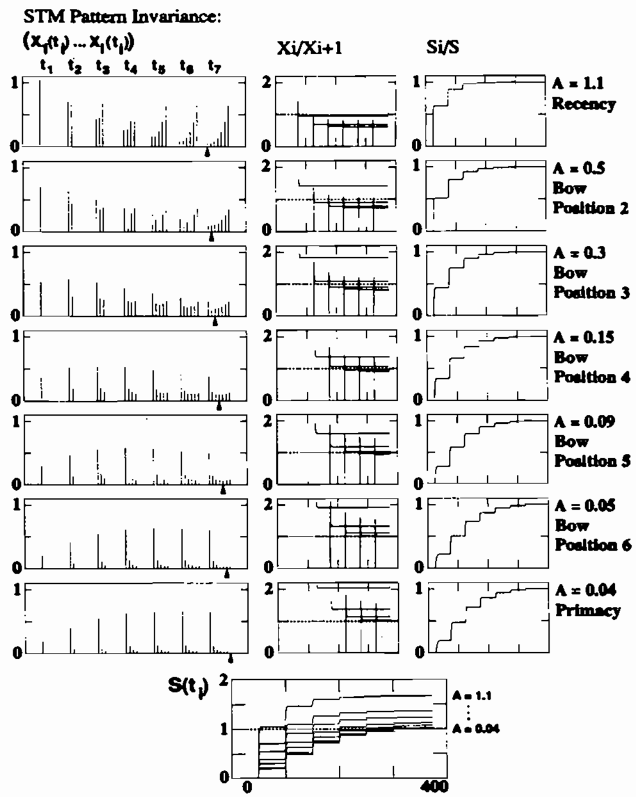

Figure 2: STORE model simulations for decreasing values of the input parameter A. The STM patterns $[x_1(t_i), \ldots, x_i(t_i)]$ show recency for large A, bowing for intermediate values of A, and primacy for small values of A. Total activity $x(t_i) \equiv S_i$ grows toward the asymptote S as i increases. When a new input I_i is stored, the previous pattern vector $(x_1, \ldots, x_{i-1})$ is amplified if $S_i \equiv x(t_i) < 1$, or depressed if $S_i \equiv x(t_i) > 1$; but the pattern of relative activities is preserved. For these simulations, input durations α_i were varied randomly between 10 and 40, with the intervals $(t_i - t_{i-1})$ set equal to 50.

similar these 2-D views must be in order to activate the same category node J.

Seibert and Waxman have successfully applied their system to the recognition of real 3-D objects. As the object moves with respect to the camera, a temporally ordered sequence $J_1, J_2, \ldots, J_m$ of 2-D category nodes is activated. These nodes and their transitions implicitly represent invariant 3-D properties of the object, in much the same manner as an "aspect graph" (Koenderink and van Doorn 1979). The Seibert and Waxman model learns to respond to temporal sequences of 2-D category activations with correct outputs from 3-D Object Nodes. To accomplish this, Seibert and Waxman modeled an Aspect Network that represents all the possible pairwise transitions between 2-D aspect nodes (Fig. 3a). The Aspect Network contains distinct locations N_{ij} at which sequential activation of nodes J_i and J_j are detected. The detection process at N_{ij} multiplies the activities x_i and x_j of the nodes J_i and J_j. As these activities wax and wane through time, a large product $x_i x_j$ denotes that a transition has recently occurred between the 2-D aspect nodes J_i and J_j.

The activation pattern across all the transition detectors N_{ij} forms the input to a competitive learning network (Fig. 3b). The output nodes of this network are called 3-D Object Nodes because the network learns to fire such a node only when an unambiguous sequence of 2-D aspect transitions is activated. An important feature of this model is its ability to recognize novel sequences composed of previously learned transitions. This approach to synthesizing 3-D recognition from combinations of distinct 2-D views is consistent with data of Perrett *et al.* (1987) about cells in temporal cortex that are sensitive to different 2-D views of a face.

Despite its many appealing features, the Seibert and Waxman model could face two types of limitations if used in a more general context: proliferation of connections and sensitivity to input timing. As in all networks that explicitly compute pairwise or higher order correlations, proliferation of connections may occur using Aspect Graphs, although this problem did not occur in the application considered by Seibert and Waxman. In general, each different temporal order would use a different Aspect Network to compute products of the temporally overlapping STM traces of all successive input pairs at the spatial loci N_{ij} (Fig. 3a). In order to compute all possible objects that can be represented by M distinct (and nonrepeated) 2-D Aspect Nodes J_i, one needs to represent $M!$ temporal orderings by $M!$ Aspect Networks (Fig. 3b). Each Aspect Network computes $O(M^2)$ products, which require $O(M^2)$ adaptive pathways to each 3-D Object Node.

In our modified architecture, the M 2-D Aspect Nodes J_i are the item nodes of a STORE model. Thus both order and item information are represented by analog activation patterns across these M codes. As a result, only one STORE model is needed with M nodes to represent all $M!$ temporal orders, no Aspect Networks are needed, and only $O(M)$

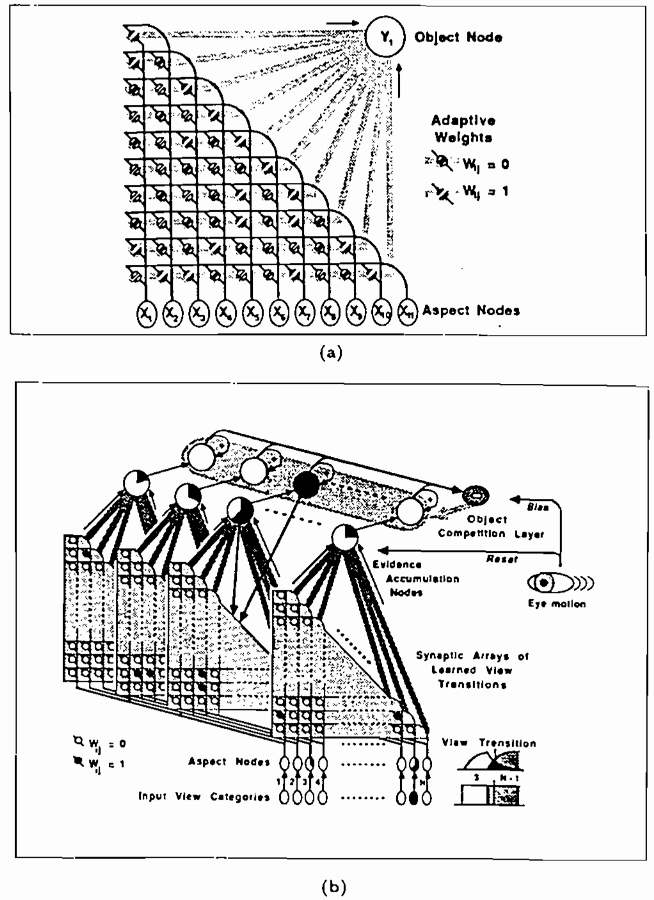

Figure 3: The Aspect Network of Seibert and Waxman detects temporal order properties by computing the temporal overlap of pairs x_i and x_j of activities at distinct locations N_{ij} and then learning the pattern of overlapping traces to code transitions between 2-D aspects. (a) A single-object Aspect Network. (b) A complete multi-object Aspect Network in which each 2-D Aspect Node fans out to contact the Aspect Networks corresponding to all 3-D Object Nodes, which compete among themselves according to winner-take-all competitive learning rules. Reprinted with permission (Seibert 1991).

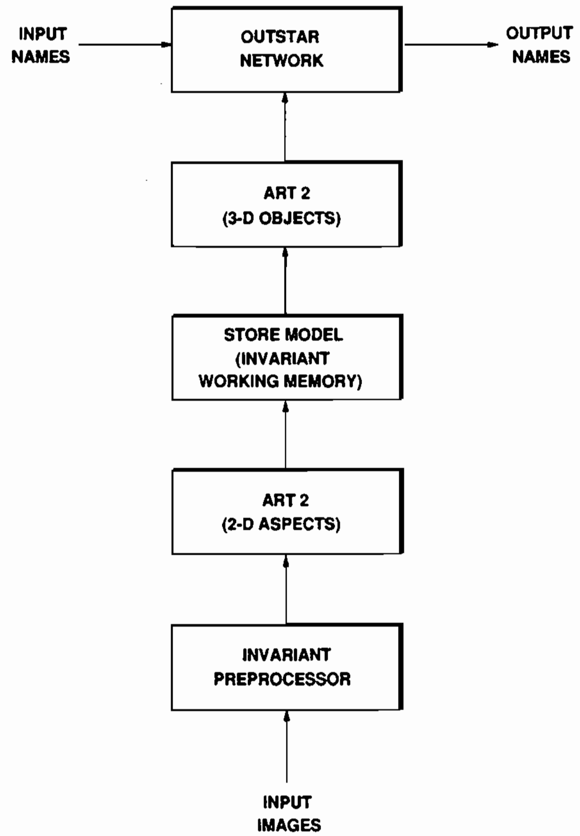

Figure 4: Processing stages of an ARTSTORE model for invariant 3-D object recognition.

adaptive pathways are needed from the STORE model to each 3-D Object Node (Fig. 4).

This substantial reduction in the number of connections is complemented by invariant temporal order properties and a simpler learning law. The Seibert and Waxman computation of aspect transitions using products of successive STM traces is sensitive to changes in input duration and interstimulus interval. They partially compensate for variations

in input duration by using a specialized LTM law in their adaptive filters whose adaptive weights converge to 1 if the corresponding product exceeds a threshold, and zero otherwise (Fig. 3a). Such an approach cannot, however, compute the order of events separated by long interstimulus intervals β_i. The working memory representation of a STORE model automatically discounts variations in input durations and interstimulus intervals. Thus the invariant temporal order code of a STORE model can directly input to a standard ART 2 network, which can automatically learn to select different 3-D Object Nodes in response to different analog patterns of temporal order information over the same fixed set of working memory item nodes. Because the model in Figure 4 joins together ART and STORE models, it is called an ARTSTORE model.

The ARTSTORE model also enables each 3-D Object Node to learn an arbitrary output pattern via outstar learning (Grossberg 1968, 1978b). To accomplish this, each 3-D Object Node is the source cell of an outstar (Fig. 4). All the outstars converge on the same outstar border where an output name can be represented in an arbitrary format by an external teacher. Thus, in response to a 3-D object moving with respect to the Invariant Preprocessor, the architecture outputs an object name when enough information about the object's 2-D aspects and their temporal order have accumulated. The total self-organizing system uses the following cascade of processing stages: Invariant Preprocessor → ART 2 (2-D Aspects) → STORE model (Invariant Working Memory) → ART 2 (3-D Objects) → (Outstar Network). This is a self-organizing multilevel instar–outstar map specialized for invariant 3-D object recognition (Carpenter and Grossberg 1991).

5 Control of Working Memory and Temporal Learning

Reset of the working memory can be autonomously controlled by the object tracking system that Seibert and Waxman have incorporated into their Invariant Preprocessor. This system enables the architecture's camera to continuously track a moving object. As continuous tracking occurs, a sequence of 2-D aspects is learned and encoded in working memory, after which a ballistic camera movement focuses on a new object. We assume that working memory is reset, and thereby cleared, when a ballistic movement occurs; for example, by reducing the gain of the recurrent interactions between the variables y_i and x_i in the STORE model. As a result, each sequence of simultaneously stored 2-D aspects represents the same 3-D object with high likelihood.

ART 2 learning of each working memory pattern may be controlled in either of two ways: (1) Unsupervised learning: Here each new entry into working memory causes ART 2 to choose and learn a new category. Each subsequence (J_1), (J_1, J_2), (J_1, J_2, J_3), ... of 2-D aspect nodes can then learn to activate its own ART 2 node. Only those subsequences which are associated with names of 3-D objects generate output predictions. (2) Supervised learning: Here an ART 2 learning gate is opened only when a teaching input to an outstar occurs. Consequently, only those sequences $(J_1, J_2, \ldots)$ that generate 3-D object predictions will learn to activate ART 2 categories and their outstar predictions. The number of learned ART 2 categories is hereby minimized. In either case, the ART 2 module can learn to select those combinations of item and order information that are predictive of an object by using its top-down expectation and vigilance properties (Carpenter and Grossberg 1987).

6 Concluding Remarks

The present model illustrates how a hierarchically organized neural architecture can self-organize a higher order type of invariant recognition by cascading together a combination of self-organizing modules, each of which computes a simpler invariant property. The Invariant Preprocessor computes a position/size/rotation invariant; the first ART 2 computes a self-calibrating similarity invariant of 2-D aspects; the STORE model computes a temporal order invariant; and the second ART 2 computes a self-calibrating similarity invariant of 3-D objects. In particular, the self-calibrating similarity invariant of 2-D aspects needs the temporal invariance of working memory to gain full effectiveness. This is so because the timing of individual outputs from the 2-D aspect nodes can depend in a complex way on the 3-D shape of an object and its relative motion with respect to the camera or other observer.

Acknowledgments

The authors wish to thank Kelly Dumont, Diana Meyers, and Carol Yananakis Jefferson for their valuable assistance in the preparation of the manuscript. G. B. was supported by DARPA (AFOSR 90-0083). G. A. C. was supported in part by British Petroleum (89-A-1204), DARPA (AFOSR 90-0083), and the National Science Foundation (NSF IRI 90-00530). S. G. was supported in part by the Air Force Office of Scientific Research (AFOSR 90-128, AFOSR 90-0175), DARPA (AFOSR 90-0083), and the National Science Foundation (NSF IRI 90-24877). This is Technical Report CAS/CNS-TR-91-014, Boston University.

References

Atkinson, R. C., and Shiffrin, R. M. 1971. The control of short term memory. *Sci. Am.* August, 82–90.

Baddeley, A. D. 1986. *Working Memory*. Clarendon Press, Oxford.

Carpenter, G. A., and Grossberg, S. 1987. ART 2: Self-organization of stable category recognition codes for analog input patterns. *Appl. Opt.* **26**, 4919–4930.

Carpenter, G. A., and Grossberg, S. (eds.) 1991. *Pattern Recognition by Self-Organizing Neural Networks*. The MIT Press, Cambridge, MA.

Cohen, M. A., Grossberg, S., and Stork, D. 1987. Recent developments in a neural model of real-time speech analysis and synthesis. In *Proceedings of the IEEE International Conference on Neural Networks, IV, San Diego*, M. Caudill and C. Butler (eds.), pp. 443–454. IEEE, Piscataway, NJ.

Cohen, M. A., and Grossberg, S. 1987. Masking fields: A massively parallel neural architecture for learning, recognizing, and predicting multiple grouping of patterned data. *Applied Optics* **26**, 1866–1891.

Elman, J. L. 1990. Finding structure in time. *Cognitive Science*, **14**, 179–211.

Grossberg, S. 1968. Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity. *Proc. Natl. Acad. Sci. U.S.A.* **59**, 368–372.

Grossberg, S. 1970. Some networks that can learn, remember, and reproduce any number of complicated space-time patterns, II. *Studies Appl. Math.* **49**, 135–166.

Grossberg, S. 1973. Contour enhancement, short-term memory and constancies in reverberating neural networks. *Studies Appl. Math.* **52**, 217-257.

Grossberg, S. 1976. Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. *Biolog. Cybernet.* **23**, 121–134.

Grossberg, S. 1978a. Behavioral contrast in short-term memory: Serial binary memory models or parallel continuous memory models? *J. Math. Psychol.* **17**, 199–219.

Grossberg, S. 1978b. A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. In *Progress in Theoretical Biology*, Vol. 5, R. Rosen and F. Snell (eds.), pp. 233–374. Academic Press, New York. Reprinted in Grossberg, S. (ed.) 1982. *Studies of Mind and Brain*. Reidel Press, Boston.

Grossberg, S., and Kuperstein, M. 1989. *Neural Dynamics of Sensory-Motor Control*. Pergamon, Elmsford, NY.

Grossberg, S., and Pepe, J. 1971. Spiking threshold and overarousal effects in serial learning. *J. Stat. Phys.* **3**, 95–125.

Grossberg, S., and Stone, G. O. 1986. Neural dynamics of attention switching and temporal order information in short term memory. *Memory Cog.* **14**, 451–468.

Gutfreund, H., and Mezard, M. 1988. Processing of temporal sequences in neural networks. *Physiol. Rev. Lett.* **61**, 235–238.

Guyon, I., Personnaz, L., Nadal, J. P., and Dreyfus, G. 1988. Storage retrieval of complex sequences in neural networks. *Physiol. Rev. A* **38**, 6365–6372.

Healy, A. F. 1975. Separating item from order information in short-term memory. *J. Verbal Learn. Verbal Behav.* **13**, 644–655.

Jordan, M. I. 1986. Serial order: A parallel distributed processing approach. Institute for Cognitive Science, Report 8604. University of California, San Diego.

Koenderink, J. J., and van Doorn, A. J. 1979. The internal representation of solid shape with respect to vision. *Biol. Cybernet.* **32**, 211–216.

Lee, C., and Estes, W. K. 1981. Item and order information in short-term memory: Evidence for multilevel perturbation processes. *J. Exp. Psychol.: Human Learn. Memory* **1**, 149–169.

Perrett, D. I., Mistlin, A. J., and Chitty, A. J. 1987. Visual neurones responsive to faces. *Trends Neurosci.* **10**, 358–364.

Ratcliff, R. 1978. A theory of memory retrieval. *Psychol. Rev.* **85**, 59–108.

Reeves, A., and Sperling, G. 1986. Attention gating in short-term visual memory. *Psychol. Rev.* **93**, 180–206.

Schreter, Z., and Pfeifer, R. 1989. Short-term memory/long-term memory interactions in connectionist simulations of psychological experiments on list learning. In *Neural Networks from Models to Applications*, L. Personnaz and G. Dreyfus (eds.). I.D.S.E.T., Paris.

Seibert, M. C. 1991. Neural networks for machine vision: Learning three-dimensional object recognition. Boston University, Ph.D. Thesis.

Seibert, M. C., and Waxman, A. M. 1990a. Learning aspect graph representations from view sequences. In *Advances in Neural Information Processing Systems 2*, D. S. Touretzky (ed.), pp. 258–265. Morgan Kaufmann, San Mateo, CA.

Seibert, M. C., and Waxman, A. M. 1990b. Learning aspect graph representations of 3D objects in a neural network. In *Proceedings of IJCNN-90, Washington, D.C.*, Vol. 2, M. Caudill (ed.), pp. 233–236. Erlbaum, Hillsdale, NJ.

Seibert, M. C., and Waxman, A. M. 1992. Learning and recognizing 3D objects from multiple views in a neural system. In *Neural Networks for Perception*, Vol. 1, H. Wechsler (ed.). Academic Press, New York, pp. 426–444.

Wang, D., and Arbib, M. A. 1990. Complex temporal sequence learning based on short-term memory. *Proc. IEEE* 78(9), 1536–1543.

Received 22 April 1991; accepted 2 October 1991.