PERGAMON

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Vision Research 43 (2003) 2669–2695

Vision Research

www.elsevier.com/locate/visres

Sequential memory-guided saccades and target selection: a neural model of the frontal eye fields

Jude F. Mitchell [*], David Zipser

Cognitive Science, University of California at San Diego, La Jolla, CA 92093-0515, USA

Received 25 April 2002; received in revised form 8 April 2003

Abstract

We present a neural model of the frontal eye fields. It consists of several retinotopic arrays of neuron-like units that are recurrently connected. The network is trained to make memory-guided saccades to sequentially flashed targets that appear at arbitrary locations. This task is interesting because the large number of possible sequences does not permit a pre-learned response. Instead locations and their priority must be maintained in active working memory. The network learns to perform the task. Surprisingly, after training it can also select targets in visual search tasks. When targets are shown in parallel it chooses them according to their salience. Its search behavior is comparable to that of humans. It exhibits saccadic averaging, increased reaction times with more distractors, latency vs accuracy trade-offs, and inhibition of return. Analysis of the network shows that it operates like a queue, storing the potential targets in sequence for later execution. A small number of unit types are sufficient to encode this information, but the manner of coding is non-obvious. Units respond to multiple targets similar to quasi-visual cells recently studied [Exp. Brain Res. 130 (2000) 433]. Predictions are made that can be experimentally tested.

Keywords: Neural model; Frontal eye fields; Working memory; Visual search

1. Introduction

Much work in attention and eye movement planning focuses on how a single visual item is selected from several, i.e. visual search. However neither attention nor eye movement planning is limited to such unitary selection. We often need to keep several items in mind simultaneously, shifting attention back and forth between them. Attention can be distributed among several items, for example, when tracking multiple objects through a scene (Pylyshyn & Storm, 1988; Sears & Pylyshyn, 2000). Likewise, in planning eye movements it appears that multiple targets can be processed concurrently (Becker & Jurgens, 1979; McPeek & Keller, 2001). The ability to divide and shift attention is important in many natural tasks. However, little is known about the neural mechanisms involved.

One task that requires several items be kept in mind is the sequential saccade task. In this task a sequence of targets is flashed and then disappears. Subjects maintain the items in working memory, and later when cued, move their eyes to each in the order they were presented.

Depending on the details of the task, different attentional and memory resources are necessary to solve it. If the task involves only a small set of highly practiced sequences, or blocks in which the same sequence is repeated, the resources needed can be low, i.e. motor sequences can be learned and retrieved from procedural memory. Much is already known about the neural mechanisms involved in this kind of task for reaching (Shima & Tanji, 2000; Tanji & Shima, 1994) and for eye movements (Isoda & Tanji, 2002; Lu, Matsuzawa, & Hikosaka, 2002). If instead there is a larger range of target locations and thus many possible sequences, and if the sequences are randomly interleaved, then a procedural response becomes difficult. Then targets must be actively maintained in working memory for saccades to be planned to them.

We are interested in the working memory version of the task. To solve the task, the brain must maintain two types of information: (1) the location of multiple targets, and (2) their priority.

[*] Corresponding author.
E-mail addresses: jude@salk.edu (J.F. Mitchell), zipser@cogsci.ucsd.edu (D. Zipser).

0042-6989/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0042-6989(03)00468-1

Recent lesion studies indicate the frontal eye fields (FEF) play a central role in this version of the task (Schiller & Chou, 2000a). Lesions of the FEF cause severe and prolonged deficits in maintaining the order of targets. The supplementary eye fields (SEF) is also important in sequential saccades (Pierrot-Deseilligny, Rivaud, Gaymard, Muri, & Vermersch, 1995). However, the SEF appears to play a less important role in this specific task. SEF lesions cause only partial deficits that recover (Schiller & Chou, 2000a; Sommer & Tehovnik, 1999).

A small number of studies have investigated the neural mechanisms involved in the working memory version of the task. All of them have recorded from cells in the FEF. Two classes of cell are found that encode a memory of target locations. Both have local receptive fields and code in a retinal coordinate frame. The first class, the quasi-visual cells (also called visual tonic cells), respond to any potential target that falls inside their receptive field (Tian, Schlag, & Schlag-Rey, 2000). The second class, visuo-movement cells, respond selectively for the next intended target (Segraves, 1992; Segraves & Goldberg, 1987). It remains unclear how these cells encode the order of targets, or how the next target gets selected at each step in a sequence.

One computational solution is a queue. In a queue several items are stored separately and in sequence. The first item to enter the queue is the first processed. After one item is done, then others shift forward and are processed. This continues until the queue is empty. This is an effective strategy for keeping multiple items in memory, prioritizing them, and then shifting attention between them.

Xing and Andersen (2000) presented a neural network model that implements a queue. A separate group of neuron-like units maintains information for each target stored in memory. The groups are organized in order of the targets' priority in the queue. The first group stores the location of the first target. The second stores the location of the second target, and so on. After a saccade is finished for the first target, its location is cleared from memory in the first group and feed-forward excitation from the second group loads the target waiting behind it. In this manner, all of the targets waiting in the queue shift forward until a saccade has been made to each of them.

The recent physiological findings from the FEF do not support a queue (Tian et al., 2000). While visuo-movement cells are selective to the next target, no class of cells is found that is selective to only the second target, or to only the third target, etc. Instead, there is a class of cells, the quasi-visual cells, which responds for any target. This suggests that another strategy may be involved.

Here we present a novel strategy. It is implemented in a neural network model. The model was generated using neural systems identification (Zipser, 1992). Neural systems identification requires specifying the temporal sequence of inputs and outputs for the task rather than the detailed connectivity of the network. An optimization, or 'learning', procedure configures the network to implement the task.

The network model was developed in stages. First it was trained to perform a simplified version of the saccade task. The targets are flashed in sequence and after a delay the network outputs their locations in the order presented but without moving the eyes to them. It is as if saccade commands were generated, but the eyes were paralyzed. Although simple, this task is sufficient to teach the network how to encode multiple target locations and their order in working memory.

After this initial phase of training, we revise the network to perform the task with eye movements. Eye movements introduce an additional problem of updating. With each move of the eyes, the locations of secondary targets on the retina also move. If the targets were visible these changes would be registered directly in the visual input. For remembered targets these changes must be generated internally. We extend the network to perform this updating with mechanisms already proposed in the literature. The revised network becomes more realistic, both in terms of its behavior and its physiological properties, but still uses the same working memory mechanism.

The network architecture is designed to resemble the FEF. We model the FEF for two reasons. First, a recent study has recorded from single units in the FEF during the working memory version of the task (Tian et al., 2000). It provides data necessary to constrain our model. Second, although the SEF play an important role in sequential saccades, their role seems secondary for this particular task (Schiller & Chou, 2000a; Sommer & Tehovnik, 1999).

Although we focus on the FEF, the mechanisms identified could be relevant to other areas. In fact this seems likely given that most saccade areas share cell types with similar physiological properties and are highly interconnected (Chafee & Goldman-Rakic, 1998; Pare & Wurtz, 2001; Sommer & Wurtz, 2001); for review (Wurtz, Sommer, Pare, & Ferraina, 2001).

The model network consists of four arrays of neuron-like units. The arrays are retinotopic, shift-invariant (Fukushima, 1980), and fully connected. Together they form a large recurrent network. Two of these arrays act as the network's *outputs*. They are trained to behave like the cell classes found in the FEF. They have local receptive fields and code in the retinal coordinate frame. One array has units that behave like saccadic burst cells. They decide where to move the eyes. The second array has units that behave like quasi-visual cells. They encode a memory for target locations over delay periods. The other two arrays in the network are *hidden*. They are not trained to perform any specific behavior. Their properties emerge as the network learns to solve the simplified task.

The input to the network consists of a retinal array and a fixation signal. The retinal array contains units that resemble phasic visual cells. They give the flashed targets in sequence. The fixation signal controls when saccadic commands are made. It suppresses the activity of the saccadic burst units until a saccade is desired. During training it is set by an algorithm, but later it is replaced with a more realistic signal that is part of the network's internal dynamics.

One desirable feature of the architecture is that despite its size, it remains relatively simple. In total, it contains 4 arrays of 32 units with 19,460 synaptic connections. However, it can be specified by only 100 parameters. The reduction in complexity is due to two constraints on the architecture. First, each array is shift-invariant. This means that each unit in an array shares the same weight pattern with the only difference being that the pattern is shifted to its corresponding retinal location. Second, the weight pattern connecting any two arrays is constrained to be a difference of Gaussians (DOG) function. This function gives a reasonable approximation to the winner-take-all interactions that have been posited to occur between cells at different retinal locations in saccade areas (Schlag, Dassonville, & Schlag-Rey, 1998; Munoz & Istvan, 1998). The function has 5 parameters. Thus the connectivity between any two arrays requires only 5 parameters. The connectivity between all 4 arrays plus the inputs requires 100 parameters to describe. During learning, it is these parameters which are optimized instead of the individual weighted connections.

The network resulting from training is what we call a *distributed queue*. It represents the same information found in a queue, but its units do not segregate into groups that selectively encode for one target. Instead, each responds to several targets in a distributed manner that resembles quasi-visual and the visuo-movement cells of the FEF. Predictions are made that can be experimentally tested.

One surprising result is that when targets are presented in parallel, the network selects them according to their salience, i.e., it performs target selection in visual search. It makes a saccade to the brightest target first, the second brightest second, and so on. We compare the behavior of the model to humans in search. It exhibits spatial averaging, increased reaction times with more distractors, latency vs. accuracy trade-offs, and inhibition of return.

2. The saccade task

The task resembles the triple step paradigm (Tian et al., 2000). It begins by flashing three saccade targets in sequence at different locations. The targets appear for a short time and then disappear. Following a delay, a fixation cue turns off and the location of each target is output in the order they were presented. The output is a purely abstract representation of a saccade at this point. It specifies a command for where to make a saccade similar to burst cells, but does not actually move the eyes. Later we revise the model so eye movements are made and the retinal location of secondary targets is updated to account for the corresponding changes in eye position.

The task is represented to the network by a temporal sequence of inputs and the corresponding outputs. The inputs consist of a 1-D retinal array of visual units and a fixation unit. The visual units give the flashed targets in sequence. The fixation unit signals when to make saccades. The outputs consist of two 1-D arrays. One output array contains saccadic burst units that represent the desired movements. The other contains memory units that are trained to encode each target location from the time it appears until the time it is output by the saccade units.

The activity of the inputs and desired saccade outputs is shown through a typical trial in Fig. 1. The inputs and outputs are shown as a column of squares at each time step (time on the horizontal axis). Each square represents a unit with its size being proportional to the unit's activation (zero activity being a small square). The 1-D retinal arrays are each shown with 8 units to ease illustration. In the actual task there are 32 units in each array and active locations are Gaussian distributed ($\sigma = 2$ units). This makes it possible to represent a continuous range of locations. Details for generating trials and computing the input–output sequences are given in Section 13.

The 1-D input array of visual units encodes a sequence of flashed targets. Each target appears at a different retinal location and remains visible for three time steps. The units have Gaussian receptive fields ($\sigma = 2$ units). Their temporal response to stimuli is phasic. It decays exponentially from the time of stimulus onset and then completely turns off after three time steps (Fig. 1 inset). This behavior is roughly consistent with phasic visual cells that are found in the FEF (Thompson, Hanes, Bichot, & Schall, 1996). Since no trace of the targets remains in the input, the network must learn to remember what it has seen.

The fixation input tells the network when to make a saccade. It is designed to be similar to fixation cells found in the FEF. They maintain a high level activity during fixation which then drops at the time of saccades (Everling, Dorris, Klein, & Munoz, 1999; Everling & Munoz, 2000; Hanes, Patterson, & Schall, 1998; Munoz & Wurtz, 1993a). Our fixation unit has a high activation of 1 during delays that then drops to 0 for saccades (bottom row Fig. 1). The first time it drops, the saccade outputs should choose the first target. The second time it

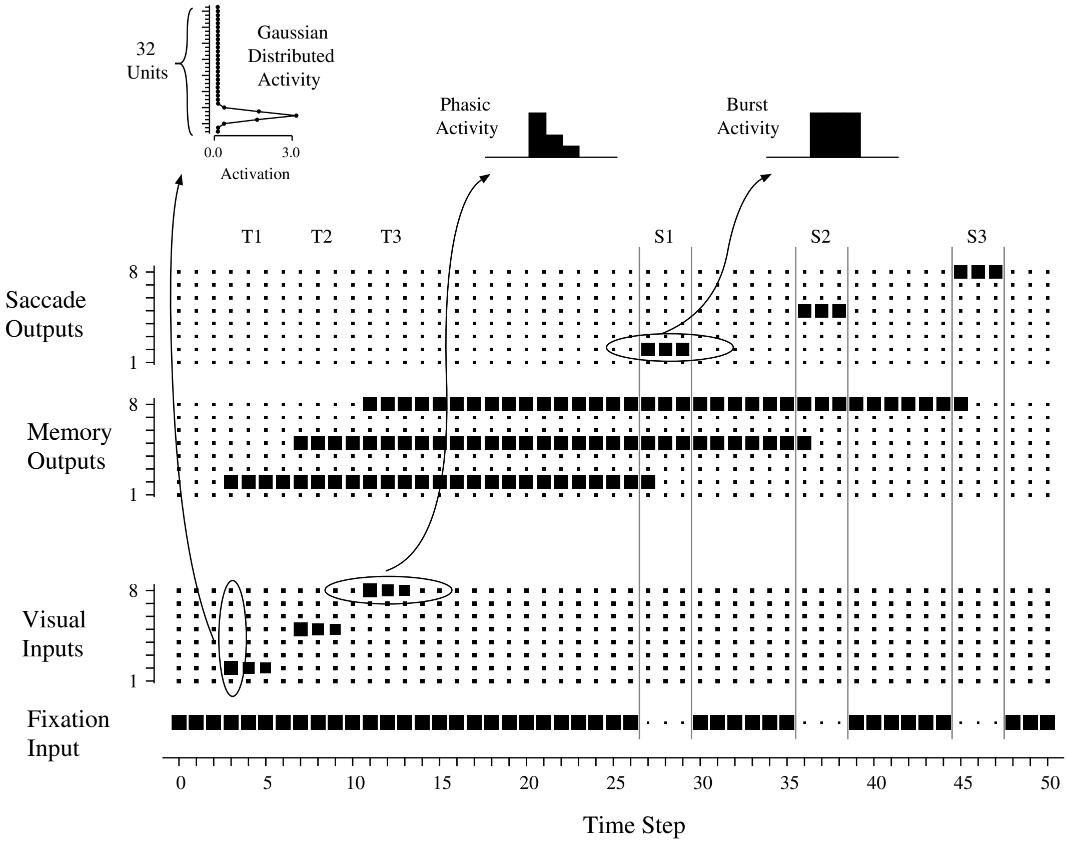

Fig. 1. A sequential task trial. Targets are flashed in sequence (T1, T2, T3) and following a delay saccades are made to each in order (S1, S2, S3). Time is on the horizontal axis. Each time step is about 20 ms. The network inputs and outputs are depicted vertically at each time step. Each unit is shown as a square with the size proportional to its activation. The 1-D arrays are depicted here with 8 units. In the real task there are 32 and active locations are distributed with a Gaussian shape.

drops, they should choose the second target, and so on. Although this control of fixation is artificial, it is important to the training process. By specifying when to make a saccade, we know when to train the saccade outputs. Once training is done, we can relax this constraint and design a more realistic signal that is part of the network's internal dynamics.

Saccade outputs consist of a 1-D array similar to the visual units. But instead of encoding the presence of targets, they encode a motor command to move the eyes to a retinal location. Like visual units, they have Gaussian receptive fields ($\sigma = 2$). They are silent until the time of a saccade is desired. Then they output a burst of activity at the target location. This is consistent with how burst cells in the superior colliculus and FEF encode movement commands (Goldberg & Bruce, 1990; Sparks, 1989).

The duration of the saccadic bursts lasts three time steps. This is analogous to about 60 ms, which is comparable to the duration of movements in the triple-step experiments (Tian et al., 2000). We do not model the detailed temporal profile of the burst activity. The outputs maintain a constant level of heightened activity over the movement (Fig. 1 inset).

The memory outputs encode the locations of remembered targets. They are necessary to teach the network to sustain a memory through long delay periods. This strategy has been used previously (Moody, Wise, Pellegrino, & Zipser, 1998). They maintain a constant level of activity for each target from the time it appears until it is output for a saccade. They have Gaussian receptive fields similar to the visual units. Their behavior is consistent with what is currently known about the activity of quasi-visual cells (Tian et al., 2000).

3. The neural network

The network architecture consists of four 1-D arrays of neuron-like units that are retinotopically organized. Each array is *shift-invariant*. This means that although it contains many units, each of them is identical except for a shift in their retinal location.

The four arrays are fully connected to form a large recurrent network. The architecture is depicted in Fig. 2. The first array, Y_1, is the saccade output. The second, Y_2, is the memory output. The other two, Y_3 and Y_4, are hidden.

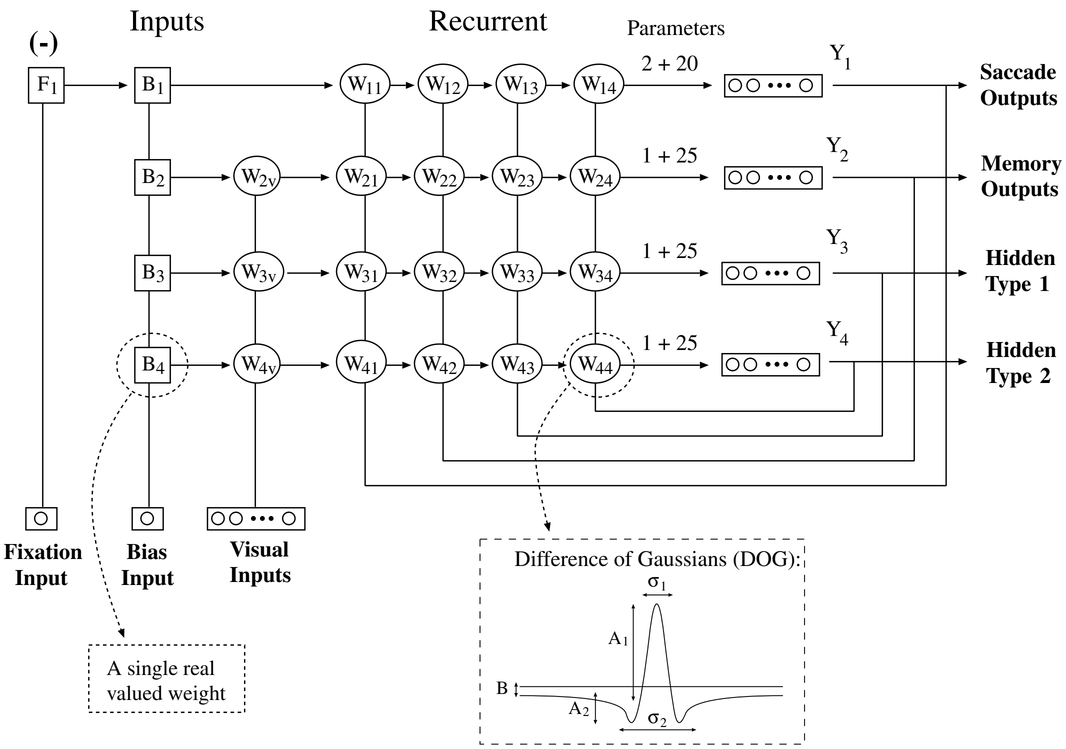

Fig. 2. Model architecture. The network consists of four 1-D arrays of units that are recurrently connected and retinotopically organized (Y_1–Y_4). The arrays are all shift-invariant. This means that the units in each array share identical properties except for their retinal location. The connectivity between each pair of arrays is described by a 1-D weight pattern. Each unit shares the same pattern, but it is shifted to the corresponding retinal location. The pattern connecting any two arrays is described by a difference of Gaussians (DOG) function with five parameters (B, A_1, σ_1, A_2, and σ_2). An example is shown for the connection W_{44}. Connection nodes that represent DOG patterns are drawn as circles. The input to the network comes from an array of visual units and two single lines for a fixation and a bias unit. Connections from the visual array are also defined by DOG patterns. The connections from the fixation and the bias input are single valued weights. For them, the same weight connects to the entire array. Those nodes are drawn as squares.

The number of hidden arrays can be varied parametrically. In unconstrained neural networks it is well known that given a sufficient number of hidden units any well-defined task can be solved (Hornik, Stinchcombe, & White, 1989). The situation is somewhat different in our architecture because each of the units in an array is constrained to be the same type, only differing in retinal location. For this case, any problem can in principle be solved if there are enough hidden arrays, i.e., enough types. We found that two were enough to solve the sequential saccade task as defined.

The input to the network comes from a retinal array of units and from two single lines for a fixation and a bias input. The retinal array of units gives the flashed targets in sequence. It connects to the memory outputs and hidden arrays, but not the saccade outputs. The saccade outputs do not receive direct visual input. The fixation input tells the network when saccades should occur. It connects to the saccade outputs which represent the saccades. The bias input connects to each array. It has a constant activity of 1. It allows each array to adjust its baseline level of activity.

The connectivity of the saccade outputs is consistent with what is known about burst cells. Burst cells occur in the deep layers of cortex. They do not receive direct projections from the input layer of cortex. This detail is included in the network by omitting the connection from visual inputs. The burst cells are also known to be inhibited by fixation cells (Munoz & Wurtz, 1993b). This is incorporated by having the fixation input make a strong inhibitory connection to the saccade output, effectively suppressing bursts until the time of saccades.

The connectivity between each pair of arrays is described by a *weight kernel*. This is a pattern of weights that is shared by every unit in the array. Each unit uses it to connect itself with its corresponding retinal location in other arrays. There are several weight kernels in the model. They are depicted as circular nodes, W_{ij} or W_{iv}, in Fig. 2.

The weight kernels are constrained to be *difference of Gaussians* (DOG) functions plus a mean value. This function has been used in other shift-invariant neural models (Itti & Koch, 2000; Kopecz & Schoner, 1995) and appears to give a reasonable approximation of the recurrent connectivity in saccade areas (Munoz & Istvan, 1998; Schlag et al., 1998). It gives the weight between a pre- and post-synaptic unit as a function of the distance between them, x, as

$$W(x) = B + A_1 e^{-\frac{x^2}{2\sigma_1^2}} + A_2 e^{-\frac{x^2}{2\sigma_2^2}} \tag{1}$$

where B, A_1, A_2, σ_1, and σ_2 are its 5 parameters. A typical example is depicted in Fig. 2.

Although the network contains 128 recurrently connected units with 19,460 connections, only 100 free parameters have to be learned. During training, the 5 parameters of the weight kernels are optimized instead of the weights themselves. The number of parameters needed to describe the connectivity of each type is listed in Fig. 2. The saccade outputs need 22 parameters. They receive connections through 4 weight kernels (W_{11}–W_{14}). Each kernel has 5 parameters, thus giving $4 \times 5 = 20$ parameters. Two more parameters are needed due to the weights from the fixation and the bias input. This gives 22 in total. For the memory outputs and the hidden arrays 26 parameters are required. Each receives connections via 5 weight kernels, 4 recurrent (W_{i1}–W_{i4}) and 1 visual (W_{iv}). Thus $5 \times 5 = 25$ parameters are needed, plus one more for the weight from the bias to give a total of 26. In the entire network, there are 22 parameters for saccade outputs, and 26 parameters for the memory outputs and the two hidden arrays, thus giving $22 + 3(26) = 100$ parameters.

The activations of units at each time step depend on their membrane potentials. Current input from synapses determines the membrane potential. The net synaptic input current is computed by the dot product of the unit's synaptic weights with the corresponding pre-synaptic activations. For unit i the current at time $t + 1$ is

$$\mathrm{net}_i(t+1) = \sum_j w_{ij} y_j(t) + \sum_k w_{ik} x_k(t) \tag{2}$$

where j indexes the network units, k indexes the inputs, w_{ij} is the recurrent weight from a unit with activation y_j and w_{ik} is an input weight from an input with activation x_k.

The synaptic current accumulates charge on the unit's membrane. The potential on the membrane at time $t + 1$ is

$$u(t+1) = \lambda \mathrm{net}(t) + (1 - \lambda) u(t) \tag{3}$$

where λ ranges between 0 and 1. This is the discrete form of a continuous differential equation. The value λ is given as $\lambda = 1 - e^{-\Delta t/\tau}$ where Δt is the size of the time steps and τ is the membrane time constant. We use $\Delta t = 20$ ms and $\tau = 6$ ms.

The activation is a non-linear function of the membrane potential. It is intended to model the average firing rate of a cell. An essential feature of cortical neurons is that their average firing rate saturates for increasing input. We model this by

$$y = \begin{cases} \frac{1}{1+e^{-u}} & \text{if } u < 0 \\ 0.5\sqrt{1+u} & \text{if } u \geqslant 0 \end{cases} \tag{4}$$

where u is the membrane potential. This function is similar to the logistic function found in most neural network models, but it has a less severe squashing nonlinearity for positive values. This is consistent with recent measurements of the contrast-response functions of cortical neurons (Reynolds, Pasternak, & Desimone, 2000).

Although the network is large, the unit activations can be computed efficiently by exploiting its shift-invariance. We first eliminate the boundary conditions at the ends of each array by making it wrap around itself in a circle so the first unit is adjacent to the last. With this modification, the equations for updating the net synaptic currents can be rewritten as sums of convolutions between weight kernels and activity arrays. The convolutions can be computed faster by using the Fast Fourier Transform (FFT) (Press, Vetterling, Teukolsky, & Flannery, 1988). See Section 13 for details.

4. Network training

The connectivity of the network is optimized through gradient descent using a continuous variant of the *Back-propagation Through Time* algorithm (Pearlmutter, 1995). Error is not propagated on every time step of the task. It is only propagated when the fixation input drops and the saccade outputs are released from inhibition. Error for the memory outputs is propagated on the same cycles.

Gradient descent is performed in the space of DOG parameters. This requires several steps for each increment in gradient descent. First, the desired changes to the weight kernels and biases are computed using weight sharing (Fukushima, 1980; LeCun et al., 1989; Rumelhart, Hinton, & Williams, 1986). This insures that shift-invariance is maintained. The resulting shift-invariant weight changes are then used to compute the gradient direction for the five kernel parameters. The kernel parameters are updated, and then the new weight patterns are generated. This constrains the network to solutions in which the weight kernels are shift-invariant and defined by difference of Gaussians. Details are given in Section 13.

The choice of the initial kernel parameters is important. If poor parameters are chosen, the network learns extremely slowly or not at all. The best results occur by initializing the kernels randomly, but in a region of parameter space where winner-takes-all behavior emerges. Parameter ranges are given in Section 13.

The untrained network can select and store the first target of a sequence. This results from its winner-takes-all interactions. It does not know how to encode multiple targets, their order, or how to plan saccades to them. It learns to do these things during training.

Networks with different random initial weight kernels are trained in the task. Training occurs on a continuous stream of randomly generated trials. The mean squared error between the desired and actual outputs is tracked. Training stops when the error stops decreasing. This takes 40,000 trials on average.

We explored different learning rates and found they do not alter the solution as long as they are below some limit. A learning rate of 0.003 is used. It gives the fastest convergence.

5. Aborted trials in training

One problem encountered in training is what to do when an incorrect decision is made in the beginning or middle of a sequence. For the purpose of training, we must decide what is the desired output behavior following the error. But what is correct? If the network selects the second target first, then what is appropriate for the second target? Certainly not the second again.

Our approach is similar to what is done in training monkeys. Monkeys responses are monitored continuously, and as soon as an error occurs the trial is aborted. The monkey may continue after an error, but it is not rewarded for the last portion of a sequence unless the first part is correct.

We follow the same strategy. We monitor what is output as the trial unfolds. The network is always trained on the first target. If it gets it right, then the trial continues to the next one. Otherwise the trial stops, the network activity is reset to zero, and a new trial begins. If the network outputs all the targets without an error, then it may continue into the next trial without ever being reset. By the end of training it performs continuously with no pause between trials.

6. Network performance

To evaluate the network's performance after training, we simulate 1000 randomly generated trials and calculate the percentage correct. A trial is counted correct if the targets are chosen in the desired order and each is accurately identified. For each saccade command we require that the location of peak activity in the output array be within one unit of the desired peak.

Performance is evaluated at different duration delay periods. We tested with 4, 16, and 64 time step delays (80, 320, and 1280 ms roughly).

Networks with fewer than two hidden types could not learn the task. The best result achieved with one hidden type was 70% accuracy. This was at the shortest delay. At longer delays the performance drops still further. The network fails to remember the secondary targets and sometimes confuses their order.

Networks with two hidden types learn the sequential task perfectly. They identify each of the three targets in sequence with 100% accuracy. They are also robust to the duration of delay, performing equally well at the shortest and longest delays.

A remarkable finding is that some of the network versions with two hidden types generalize to a visual search task. The versions differ only by their random initial weights and by the random sequence of trials in training. Each of them solves the sequential task, but the ways they solve it are different. Some solutions are better suited to the visual search than others.

We test how each solution generalizes. No additional training is performed. The network weights are fixed, and then it is run in 1000 trials in which the targets appear simultaneously but have distinct intensity differences. Trials are counted correct if targets are chosen according to their intensity, beginning with the brightest.

One network generalizes to the visual search task with perfect performance. It selects the targets in order of their brightness. We focus on this network in our analysis and revisions. Other versions are considered in the discussion.

7. Analysis of memory mechanisms

Before considering revisions to the model, we first analyze the working memory solution identified in training. We show how units encode target locations and their order. The details are difficult to absorb at one glance. Therefore we break our analysis into three parts.

First we study the memory activity in the delay period. Units with memory activity exhibit different levels of selectivity to visual stimuli. Some respond to any of the presented targets, others are more selective. The selections made are compatible for the search and sequential tasks. This is what enables the network to generalize between tasks.

Second, we show the memory activity gives a distributed representation of the information in a queue. With a linear transformation, we can extract an ordered list of the target locations that are waiting to be executed.

Third, we analyze the connectivity of the network. The architecture discovered resembles a queue. We discuss how interactions between units maintain target order.

7.1. Memory activity

Together the hidden types and memory outputs constitute the working memory of the network. In this section we examine how targets are loaded and stored in working memory.

In Fig. 3, the network activity during the delay period is shown. The trial begins with three targets flashed

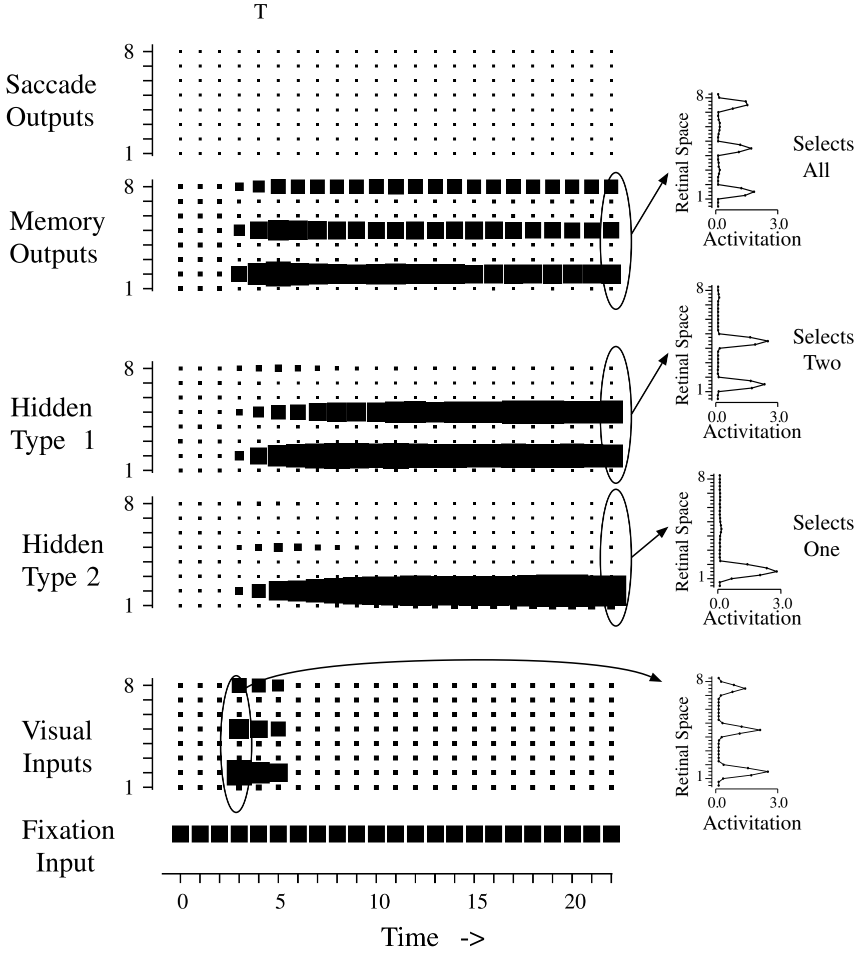

Fig. 3. Inputs, outputs, and hidden types over a visual search trial in which the targets are flashed in parallel and selected according to their brightness. Display format is the same as Fig. 1. Activity is subsampled to an array of 8 units to ease illustration (actual size is 32). Spatial cross-sections of activity are shown on the right at selected times.

simultaneously. They appear in the visual input. The brightest target is at the bottom of the 1-D retinal input, and the darkest at the top. The spatial profile of the activity is shown on the right. Visual activity is Gaussian shaped for each of the targets. Its height encodes target intensity.

Memory units sustain activity over delays after the input disappears. They respond to targets that fall inside their receptive fields. The spatial profile of response is depicted for each type on the right in Fig. 3 (space is plotted on the vertical). Units have Gaussian shaped responses similar to cells studied in the FEF (Schall, Hanes, Thompson, & King, 1995). The shape derives from the excitatory component of the difference of Gaussians weight pattern. The inhibitory component of the difference of Gaussians does not have a pronounced effect in the activity profile as it falls below threshold in the sigmoidal activation function.

Each type responds to a different number of targets. Beginning from the bottom of Fig. 3, hidden type 2 is the most selective. Its units respond only when the brightest target falls inside their receptive fields. Thus, it selects one target. Hidden type 1 responds to the first and second brightest targets. It selects two. The memory outputs respond to any of the three targets. They select all of them (as they were trained to do). Subsequently, we will refer to these different types as select-one, select-two, and select-all.

The visual inputs load targets into memory. They excite corresponding retinal locations in each of the network arrays. If the excitation is sufficiently strong, the units at that location become active. Activity can persist over delays without visual input.

What constitutes sufficient excitation is complicated. It depends on the selectivity of the unit, and also on whether or not the neighbors of that unit are already active. Some arrays respond to a maximum of one or two locations. Once they reach capacity, further visual excitation is ineffective. We examine how the recurrent connections contribute to this behavior.

Activated locations are sustained through the weight kernel connecting each array to itself. We refer to it as the *self-connecting* kernel. The other recurrent connections are unnecessary to sustain activity. Lesioning them has no significant effect. They play other roles in the network that we discuss later.

Each self-connecting pattern resembles a Mexican-hat. A few examples are shown in Fig. 4. It has excitatory connections between neighboring units and inhibitory connections between distant ones. The local excitatory connections support activity within a region in an array. Inhibitory surrounds prevent the spread of activity beyond that region. Activity can persist at a fixed location for indefinite periods.

The level of *global inhibition* between units determines the maximum number of locations that can become active in an array (Fukai & Tanaka, 1997). Global inhibition refers to the inhibitory connections between units that are distant. It depends on the bias value, *B*, in the kernel. Two kernels with different bias values are shown in Fig. 4. The top one is from the select-all units and the bottom one is from the select-one units.

When global inhibition is weak, several active locations can persist in an array. This is the case for kernel of the select-all units. It has a bias value near zero. Distant locations operate independently of each other.

Competitive dynamics emerge when global inhibition is strong. This is the case for the select-one type. Its self-connecting kernel has a large negative bias. Distant locations inhibit each other. This causes competition between locations. Only the most active can survive.

Kernels that produce a few surviving locations are also possible. They must have intermediate levels of global inhibition. This is the case for the select-two type. It is an example of *winner-takes-n* dynamics (Fukai & Tanaka, 1997). The *n* most active locations are selected.

One of the stable states of each of the memory types is silence. This is important to their operation. The competitive dynamics they implement are not strong enough to select a winning location from weak or random initial activities. If this were not the case, the network might choose target locations at random.

Target selection is determined by the visual excitation input at the start of each trial. The excitation is stronger for the brighter targets. This biases competitive dynamics in their favor. For this reason the select-one array always selects the brightest target and the select-two array selects the two brightest targets.

In the sequential task, the latency of visual excitation determines target selection. Earlier targets are favored over later ones. This is because the memory arrays accept the first target which excites them. When later targets appear they have already reached their maximum capacity and are less responsive. The select-one array thus responds to the first flashed target, but not those after it. The select-two array responds to the first two flashed targets.

The mapping between target latency and brightness is compatible. This allows the network to generalize between the sequential and search tasks.

7.2. Distributed encoding

The varying levels of selectivity found among the memory units give a distributed representation of a queue. We show how to extract an ordered list of the target locations from this activity.

The reconstructed queue is presented in Fig. 5. The input to the reconstruction is the arrays of memory activity from the delay period shown in Fig. 3. To ease illustration, the activity is rotated so space is now on the horizontal axis. The output consists of three arrays of activity: Q1, Q2, and Q3. Each is computed by a weighted sum of the input. Q1 encodes the location of

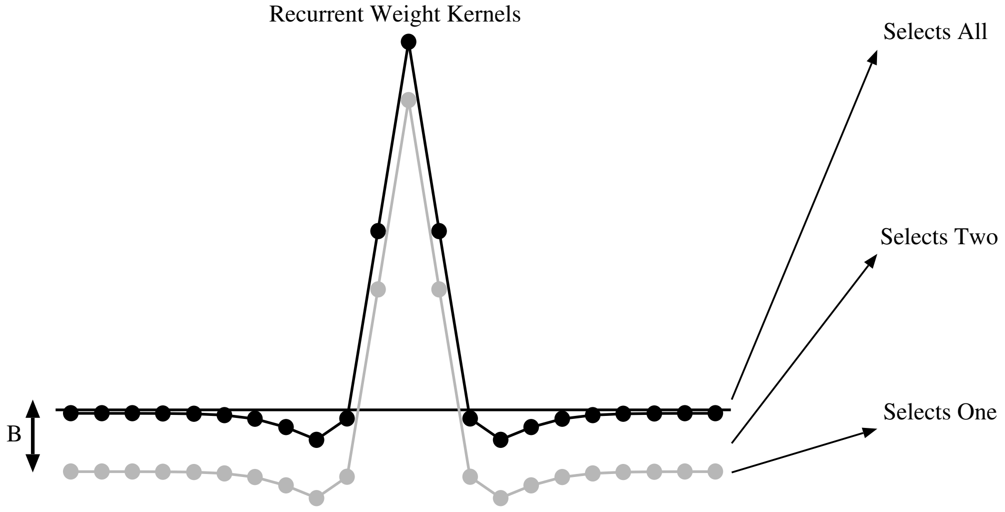

Fig. 4. Self-connecting recurrent weight kernels. Depending on the bias parameter, *B*, different numbers of locations in an array are selected and remain active. This is an example of winner-takes-*n* dynamics (Fukai & Tanaka, 1997).

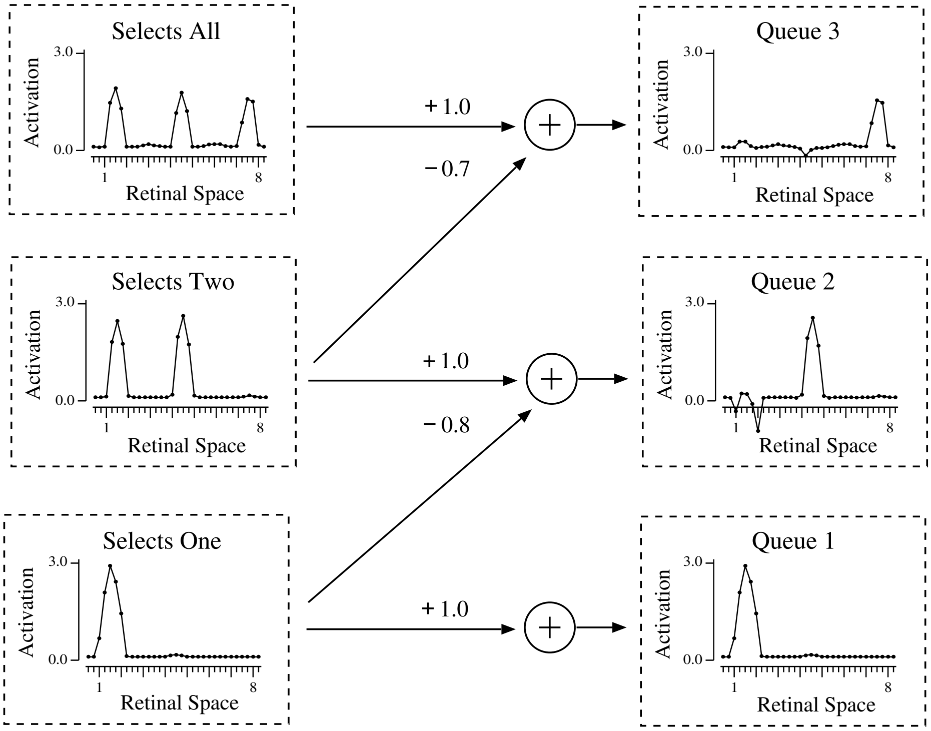

Fig. 5. Linear transformation from activity in the memory arrays (left) into activity selective to the first, second, and third target (right). A weighted superposition of array activities reconstructs each target alone.

the first target alone. Q2 encodes the second target alone, and Q3 encodes the third target.

It is trivial to determine the activity for Q1. It should be identical to the select-one array. That is because that array encodes the first target alone (Fig. 5, bottom).

Extracting the second target is more difficult because no memory type encodes it alone. The select-two units respond to it, but they also respond to the first target. The magnitude of the two responses is nearly identical. It is not possible to distinguish them on average. However, a weighted sum of select-one and select-two array activities extracts the second target. If $\vec{H}_{S1}$ is the select-one array, and $\vec{H}_{S2}$ the select-two array, then

$$\vec{Q}_2 = \vec{H}_{S2} - 0.8\vec{H}_{S1} \tag{5}$$

is the equation for a new array, $\vec{Q}_2$, which has a single peak at the location of the second target (Fig. 5, middle).

The third target can be extracted from a weighted sum of the select-all array and the select-two array. Expressing the select-all array as $\vec{H}_{SA}$ the equation

$$\vec{Q}_3 = \vec{H}_{SA} - 0.7\vec{H}_{S2} \tag{6}$$

gives a new array, $\vec{Q}_3$, that has a single peak for the third target (Fig. 5, top).

We test the accuracy of the reconstruction in 1000 randomly simulated sequential and search trials. For the parameters chosen, it always identifies the three target locations separately with perfect accuracy.

7.3. Network connectivity

The connectivity of the network can be simplified. Several of the weight kernels are near zero after training. These kernels can be removed from the network. The kernels that remain all share a similar Mexican-hat pattern. They can be simplified to have the same form, so they vary only in their magnitude and their bias value. These modifications have a negligible effect on the network's performance. The structure that remains resembles a queue.

In Fig. 6 we present a schematic of the network that remains. The order of the arrays is redrawn to reflect the flow of information. Each of the arrays is presented as a rectangular box with a plot of its population activity. Information flows from the visual input at the bottom to the burst outputs at the top. The selectivity of units increases at each ascending stage in this hierarchy. Visual units respond to all targets presented while those at the top select the single brightest target.

The connections between the arrays are drawn as lines ending with nodes. All of them share the same Mexican-hat form in which similar retinal locations are excited and distant locations inhibited. The size of the nodes reflects the magnitude of the Mexican-hat form. Exact parameters for each kernel are given in Appendix A.

The memory arrays are arranged like a queue. The select-all units represent the bottom of the queue and the

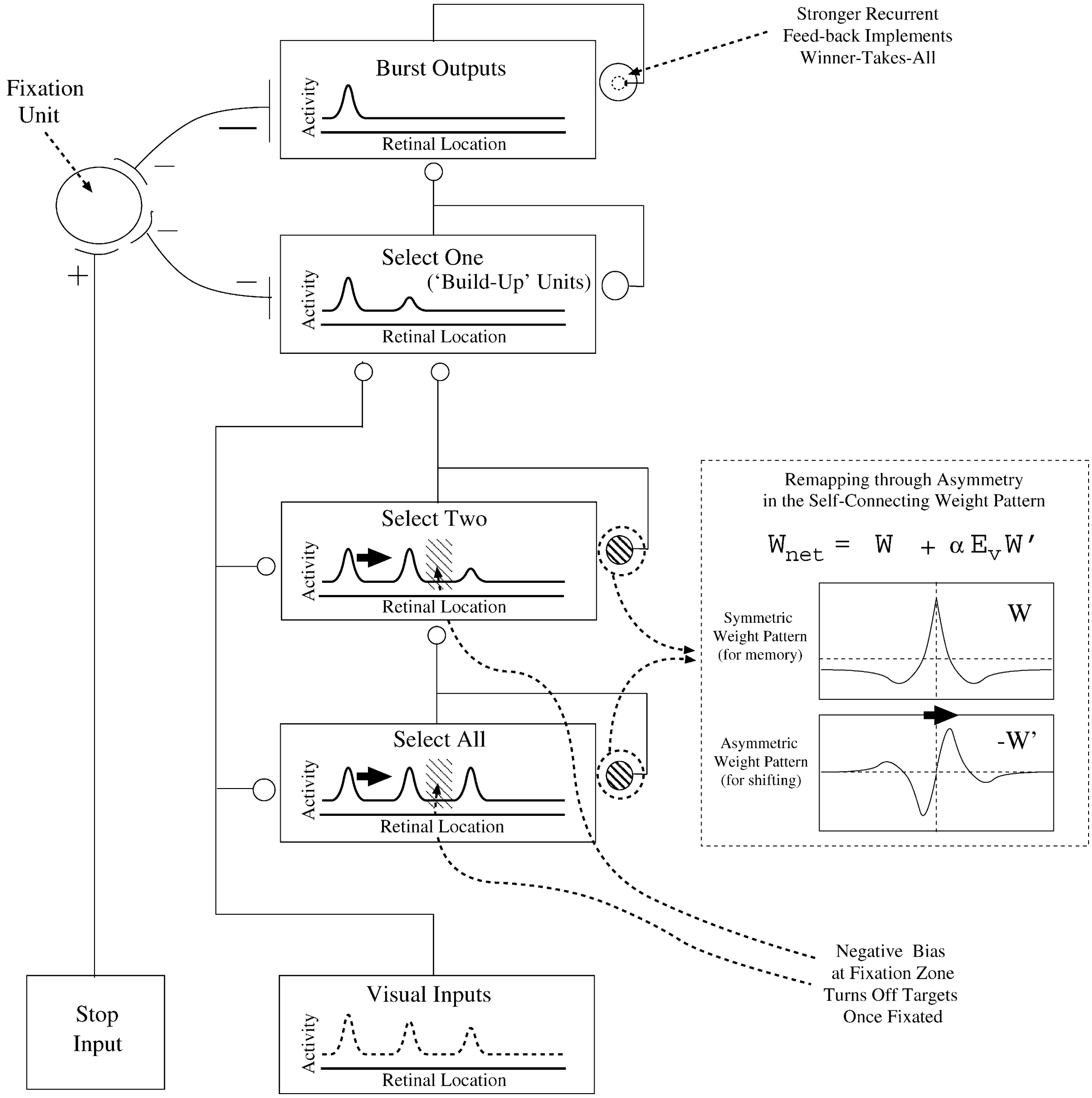

Fig. 6. Final network connectivity plus revisions. Connections near zero are removed and the network redrawn to show the flow of information. Each rectangular box represents an array of units with a schematic of their population activity. Lines terminating with a round node represent a Mexican-hat weight pattern, with the node size reflecting its magnitude. *Revisions:* On the left, a fixation unit makes reciprocal inhibitory connections to the burst array and the select-one array. The connections are spatially uniform (symbolized by termination with a line instead of a node). At the top, the self-exciting kernel of the burst outputs is enlarged to implement winner-takes-all dynamics. On the right, two revisions implement updating of remembered locations in the select two and select all arrays. First, the self-exciting kernels are modified to remap stored locations by dynamically including an asymmetric component to the weight pattern that is controlled by eye velocity, E_v. Second, extra inhibition is placed at the fixation zone of each array causing targets to turn off after being fixated.

select-one units the top. Each memory type excites the one above it. At the top of the queue, the select-one units excite the saccade outputs. They determine where the outputs will burst during saccades.

The feed-forward excitation among memory types has little effect during delay periods. It is weak relative to the self-excitation that sustains the targets already occupying a forward position. This is reflected in Fig. 6 by the larger size of the self-connecting nodes.

During saccades, the saccade target is cleared from memory and this allows remaining targets to push forward to fill its place. This continues until all targets have shifted through the queue.

The network clears targets from memory using negative feedback from the burst outputs (not shown in Fig. 6). The outputs connect to the memory arrays with an inverted Mexican-hat weight pattern. When bursts fire, they inhibit the corresponding active location in the memory arrays. This turns off the target, allowing others to shift forward and take its place. Unfortunately, this strategy does not generalize to the task with eye movements. We discuss this in more detail in the following section. An alternative strategy is proposed.

8. Post-training revisions

In this section, we extend the network to solve the problems of fixation control and updating secondary locations. Much is already known about the behavior and the physiology of these problems. Realistic models have already been proposed that employ a similar shift-invariant architecture as our model. It is relatively simple to incorporate details from those previous models. The modifications make our network more realistic, but do not alter the key features of its working memory, which is our main contribution.

There are four revisions. The first involves modifications made to both the fixation input and the burst outputs. The fixation input is replaced with a unit that is part of the network's dynamics. It interacts with the burst outputs to control saccade initiation. The burst outputs are revised in order to calibrate for this change, and also to make their behavior more realistic. The second revision involves modifications to select-two and select-all memory arrays. These arrays use eye velocity feedback to remap stored visual locations during saccades. The remapping mechanism is inspired by *moving hill* models that have been previously studied (Droulez & Berthoz, 1991; Zhang, 1996). The third revision also involves these memory arrays. The strategy for clearing old targets is revised to be compatible with the new remapping that occurs during saccades. The fourth revision adds noise to the units of the network. This enables them to exhibit stochastic behavior.

8.1. Fixation control and burst outputs

The fixation input is replaced with a fixation unit that is part of the network's dynamics. The modifications are depicted on the left side in Fig. 6. They are inspired by previous models of saccade initiation in the superior colliculus (Grossberg, Roberts, Aguilar, & Bullock, 1997; Trappenberg, Dorris, Munoz, & Klein, 2001; Wurtz & Optican, 1994). The models characterize the interaction between three classes of cell: the fixation, burst, and 'build-up' cells. Build-up cells, also called visuo-movement cells in the FEF, select the location of the next intended target immediately before saccade initiation (Schall et al., 1995). They resemble the select-one units in our network.

The activity of the fixation unit is determined by its connections to the network. It makes reciprocal inhibitory connections to the burst outputs and also to the select-one units. The connection to these arrays is uniform (i.e. the weight to each unit is the same). It has a positive bias input that gives it a high resting activity. It also receives a 'stop' input that is excitatory. This input is used to hold fixation during delays in the memory paradigm and to restore fixation when a saccade is complete. Detailed equations for its activity are given in Section 13.

Competition between the fixation and the burst and select-one units controls when saccades occur. Normally, the fixation unit has a high resting activity that completely suppresses the burst outputs. Its suppression of the select-one units, however, is much weaker. As those units become excited by visual inputs and other memory types, they are able to accumulate activity for the next desired target. The accumulation of activity initiates the saccade by inhibiting the fixation unit. As fixation turns off the burst units are released from their suppression and fire at the desired location. The burst further inhibits fixation keeping it silent through out the saccade. Fixation can not recover until the end of the saccade when they are excited by the 'stop' input turning on.

The burst units require strong recurrent feedback to support their competition with the fixation unit. We increase the magnitude of their self-exciting kernel so they have sufficient feedback (top of Fig. 6). The strength of their self-exciting connection was weak in the trained network. It was not necessary due to the artificial control from the fixation input. When fixation was on, the outputs were completely suppressed, and when it was off, they were driven in a feed-forward manner by their select-one inputs. These two modes created the desired on/off bursts in the output. In the present model, the outputs must reinforce their own activity to initiate and sustain bursts.

The revised recurrent kernel further produces more realistic winner-takes-all dynamics among the saccade outputs. These dynamics insure that a single peak of activity forms among the outputs during bursts. This is consistent with what is found experimentally. The population of burst cells fires at a single location even when there are several targets and even if the movement produced goes to a location intermediate between them (Glimcher & Sparks, 1993; van Opstal & van Gisbergen, 1990). This is a reasonable strategy given that the eye can only move to a single location at a time.

8.2. Updating secondary locations

The second revision involves updating that is necessary during saccades. Each time the eyes move during saccades, the position of targets on the retina also moves. For visible targets this change is registered by a shift on the retina. For the remembered targets it must be generated internally.

Several mechanisms have been proposed that update retinal locations during saccades (Droulez & Berthoz, 1991; Hahnloser, Douglas, Mahowald, & Hepp, 1994;

Krommenhoek, Opstal, Gielen, & Gisbergen, 1993; Mitchell & Zipser, 2001; Xing & Andersen, 2000; Zhang, 1996). We adopt one reasonable mechanism that was introduced by Zhang (1996) and is well suited to shift-invariant models like our network. The mechanism uses velocity feedback during movements to shift active locations in the memory arrays. The shifting is continuous in time and space. It resembles a 'moving hill' of activity, or in this case potentially several moving hills, that shifts gradually across a retinotopic map.

The physiological evidence for a moving hill of activity in saccade areas remains inconclusive (Soetedjo, Kaneko, & Fuchs, 2001). Early work in the cat superior colliculus demonstrated that the activity of build-up cells during saccades is consistent with a moving hill (Munoz & Guitton, 1991; Munoz, Guitton, & Pelisson, 1991). However, recent studies in the monkey indicate that the activity in both burst and build-up cells is stationary during saccades (Anderson, Keller, Gandhi, & Das, 1998; Munoz, Waitzman, & Wurtz, 1996; Soetedjo et al., 2001).

Quasi-visual cells do undergo some kind of remapping during saccades (Tian et al., 2000; Umeno & Goldberg, 2001). They begin to respond when a remembered target is brought inside their retinal receptive field. The onset of the response can precede the end of the movement. This 'remapping' keeps stored target locations spatially accurate. Whether or not it is mediated through a moving hill mechanism remains untested. We develop our model under the assumption that a moving hill is involved, but only among the quasi-visual type units. Predictions are laid out that can be experimentally tested.

The modifications for remapping are depicted on the right in Fig. 6. The select-two and select-all arrays, which encode multiple targets similar to quasi-visual cells, are altered to remap stored locations. The burst and select-one arrays are not modified. During saccades, they maintain a burst of activity at a fixed array location consistent with the recent physiological findings. It is not necessary for them to actively participate in remapping because they receive their visual input from the other memory types. As long as that information is updated, subsequent motor plans remain spatially accurate.

Remapping is mediated through the self-exciting weight pattern. The symmetric Mexican-hat pattern introduced earlier has the property that each active location remains stationary in the array. The spatial derivative of this pattern, W', is an asymmetric pattern that excites locations to one side and inhibits those on the other side. The addition of such an asymmetric component to the net weight pattern results in active locations that shift along the direction of the asymmetry. The speed and direction of the shift are controlled by adjusting the size and sign of the asymmetric component.

Eye velocity feedback, E_v, adjusts the asymmetric component in the weight pattern to give the desired remapping. The net weight pattern is given by

$$W_{net} = W + \alpha E_v W'$$

where W is the symmetric Mexican-hat weight pattern, W' is its spatial derivative, and α is a scaling constant ($\alpha = 2$). When the eye velocity is zero, the asymmetric component is zero and has no effect. Active locations remain stationary in the arrays. The asymmetric component does not become involved until a saccade occurs. In Fig. 6, the asymmetric component is depicted when a saccade to a leftward target is made. During the leftward saccade, the eye velocity takes on a negative value ($E_v = -1$) such that the asymmetry is negative ($-W'$ as shown in the bottom of the inset on the right) and active locations are driven rightward. Active locations shift to the right until the target has reached the fixation zone, at which point a stop signal is generated to end the saccade. The pattern is reversed when saccades are made to the right. The eye velocity takes on a positive value ($E_v = +1$) and active locations shift in the opposite direction.

Details for computing the eye velocity and stop signal are provided in Section 13. We assume that these signals are provided as input into the FEF. Current evidence suggest that they are encoded in brain-stem circuitry and in the cerebellum (Scudder, Kaneko, & Fuchs, 2002). The FEF receives feedback from these areas through the thalamus.

One problem with the remapping solution is that it requires that the connectivity pattern change dynamically. It is unlikely that synaptic connections between cells could change on such a fast time scale. Zhang (1996) describes a realistic implementation that is mathematically equivalent. It requires that there are a pair of arrays with identical properties. The two arrays connect to each other with the same symmetric weight pattern, but with opposite asymmetries. One is designed to push active locations to the right while the other pushes them to the left. At equilibrium, the two pairs are equally active and their asymmetries cancel out. However, during saccades their activity is modulated by eye velocity such that one becomes more active and thus gains more influence in their shared recurrent feedback. Activity shifts in the direction of the more active array. This implementation makes specific predictions about the behavior of quasi-visual units during saccades that we present later.

8.3. Clearing past targets

The third revision clears old targets from memory after saccades have been completed to them. This is an essential operation because it allows remaining targets to push forward in the queue so that subsequent saccades can be executed.

The trained network found a solution, but it was specific to the task without eye movements. When the saccade outputs fired a burst, they inhibited the corresponding retinal location in the memory arrays turning off target related activity. This strategy fails when remapping is involved because the spatial correspondence between the burst in the outputs and activity in the memory arrays breaks down. The burst in the outputs remains stationary (consistent with physiological data) while the active locations of the memory arrays shift.

An alternative strategy is to add negative bias to the memory units at the fixation zone. This is depicted at the bottom right of Fig. 6. The negative bias causes the resting activity at these central locations to be lower. During remapping, active locations can still pass through the fixation zone, but if they stop there, as is the case with the saccade target, then memory activity gradually turns off. The time necessary for turn off acts as a fixational dwell time. Once the target is cleared, others can compete to initiate subsequent saccades.

8.4. Stochastic behavior

Noise is added to units to give them probabilistic behavior. The noise is included on every time step by multiplying each unit's activation by a Gaussian random variable ($\mu = 1$, $\sigma = 0.15$). This choice is consistent with the variability found in real cells (Softky & Koch, 1996).

9. Saccades and memory updating

We examine the updating behavior of the revised network. Three events occur with each saccade performed. First, the locations of secondary locations are remapped to account for the movement of the eyes. Second, once the target is fixated, it is cleared from memory. And third, the remaining targets move forward in the queue of memory arrays in order initiate subsequent saccades.

The updating activity during a sequence of three saccades is depicted on the left in Fig. 7. The graph conventions are the same as in Fig. 3 except that the size of arrays is doubled ($N = 64$) and the integration time step has higher resolution ($\Delta t = 1$ ms). On the right, plots resembling peri-stimulus time histograms (PSTH's) are given. They show the average firing rate of a unit over the time. These plots are discussed in detail in the following section when physiological comparison are made. Here we focus on the left of the figure, which shows how the spatial profiles of activity shift and are updated among the arrays.

In the trial presented, the targets are flashed in parallel. The brightest one is at the bottom of the 1-D visual inputs, and the least bright is at the top. The saccade outputs are shown in the top row of the figure. They output bursts at the locations of each of the targets beginning with the brightest one. The time of each saccade is marked with vertical lines at the saccade onset and offset (S1, S2, S3). It is determined by the time when peak activity in the outputs crosses a threshold of 0.5.

During saccades the visual locations stored in memory are updated through a continuous remapping in the select-two and select-all arrays. This is depicted in Fig. 7 (highlighted from bottom left label). In the figure it appears like a diagonal shift of the active locations. The shifting stops once the active location corresponding to the target has reached the center of the array (i.e., the fixation zone). Once the target hits fixation, a stop signal is generated to restore fixation and suppress the burst.

After the saccade, the old target is cleared from memory in the select-two and select-all arrays. This event is highlighted in Fig. 7 (bottom middle label). The decay of memory activity is gradual, occurring over roughly 50 ms. It happens because the old target is positioned over the fixation zone which contains a more negative bias than other parts of the array. This bias is not strong enough to turn off a target that passes through fixation quickly, as can be seen earlier for the second target which passed through fixation in the first saccade. However, if a target waits at fixation, memory activity associated with it decays gradually. New targets are not selected until the old target is cleared. Thus the gradual decay provides a dwell time for fixation.

After the old target is cleared, remaining targets shift forward in the queue of memory arrays to take its place and initiate subsequent saccades. The loading of new targets is depicted in Fig. 7 for the select-one and select-two arrays (highlighted with the bottom right label). In the select-one array, the retinal location corresponding to the second target begins to become active after the first saccade. Note that the location has changed from when it first appeared in the visual inputs. This is because the select-one array receives its input from the select-two array, which has remapped locations during the first saccade. The remapping keeps saccades to secondary targets spatially accurate. Following the second saccade, the select-one units become active for the updated location of the third target. They always load the next intended target. This is consistent with the behavior of build-up cells in the FEF.

The select-two array of units also loads new targets following saccades. They encode the next two intended targets in the sequence. After the first saccade, they load into memory the third target (shown in highlighted region of Fig. 7). Before the second saccade begins, they are active to the second and third target locations.

The loading of new targets preserves the representation of order. At every stage, the build-up units select

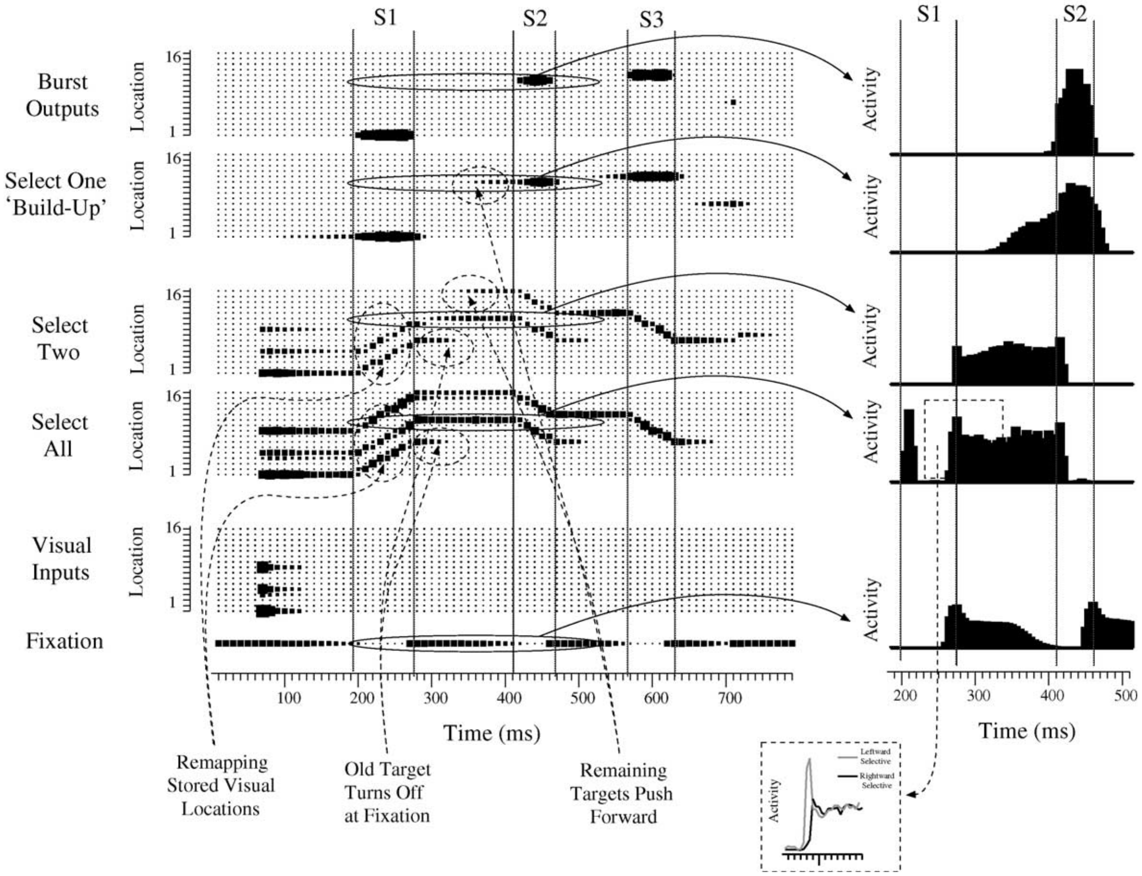

Fig. 7. Activity traces in the revised network. Plots have the same conventions as Fig. 3, with the spatial resolution of each array doubled to show more detail. The network makes three saccades (S1, S2, S3). During each saccade, the visual locations stored in the select-two and select-all memory arrays are remapped (highlighted by the bottom left label). After the saccade, the old target, which now sits at fixation, gradually decays from memory (highlighted by the bottom middle label). Then remaining targets shift forward in the hierarchy of memory arrays to take the place of the old target (highlighted by the bottom right label). This process continues with each saccade until no targets remain in memory. On the right, plots of the average firing rate over time are shown in the interval preceeding the second saccade for each type. The predictive response when a target is brought inside a quasi-visual units receptive field is highlighted on the bottom right (discussed in text).

the next target, the select-two units select the next two targets, and the select-all units encode any remaining. An ordered list of each individual target and its position in a queue can be reconstructed from this activity with the linear transformation presented previously (Fig. 4). If no target is remaining at a position in the queue, then the reconstructed activity will be uniformly low and can be distinguished by a threshold.

10. Physiological comparisons

Activity traces from single units are presented on the right of Fig. 7. Each unit resembles a class of cell that is found in the FEF. We examine each class in turn. First, the revised fixation and burst units are no longer discrete on–off signals, but instead show graded activation that is more realistic. Second, the select-two and select-all units resemble quasi-visual cells. Predictions for quasi-visual cell behavior are made that can be experimentally tested. And last, the select-one units resemble 'build-up' units which accumulate activity at the target location prior to a saccade. Saccades are initiated as the activity reaches a fixed threshold.

10.1. Fixation and burst units

The revised fixation and burst units have graded levels of activity that resemble their counter-parts in the FEF. The activity of the fixation unit is shown in the right of Fig. 7 (bottom). In the interval before a saccade it gradually decays to a value near zero, and remains near zero as the burst units fire. Then near the end of the saccade it returns with a brief transient as the stop signal turns on to terminate the saccade. Similar dynamics are observed among fixation neurons in the FEF (Everling, Pare, Dorris, & Munoz, 1998; Hanes et al., 1998). The activity of a burst unit at the target location is also shown in the right of Fig. 7 (top). It rises rapidly to a peak at the saccade onset, and then declines as the saccade finishes. Similar profiles are found in real burst cells (Hanes et al., 1998).

The graded levels of activation occur through competitive dynamics between the fixation unit and burst units. The select-one units also play a role in this competition. As they accumulate activity prior to a saccade (second trace from the top) they begin to suppress fixation. The gradual turn-off of fixation from select-one unit suppression ultimately releases the burst units to fire. We examine this behavior in more detail later.

10.2. Quasi-visual units

The select-two and select-all units resemble quasi-visual cells (also called visual tonic cells). An important feature of their activity is that they begin to respond when a remembered target is brought into their receptive field (Tian et al., 2000; Umeno & Goldberg, 2001). This response is depicted in Fig. 7 as the second target enters the receptive field (middle two traces). In physiology, this has sometimes been called a 'predictive' response because it begins slightly before the end of the saccade (Umeno & Goldberg, 1997, 2001). Since it is faster than the latency of visual input, it is taken as evidence for an internal remapping. This is consistent with the remapping that occurs within our network.

The remapping mechanism makes two specific predictions for quasi-visual cell behavior. The first is that they should exhibit a transient burst when a target passes through their receptive field during saccades. An example can be seen in the activity trace of the select-all unit (Fig. 7, second trace from bottom). At the start of the trace during the first saccade (S1) the unit responds very briefly as the third target enters and leaves its receptive field. Quasi-visual cells should have similar transients to targets that pass through their receptive field if they use a moving hill mechanism.

The second prediction is that eye velocity should modulate the predictive component of the remapping response. This prediction is specific to the implementation of the moving hill. As mentioned earlier, the direction and size of shifts are controlled by adding an asymmetric component to the self-exciting weight pattern. However, it is unlikely that this could be implemented by dynamically changing the connectivity during saccades. A realistic alternative involves a population of similar units that have different biases for shifting activity. For example, one set could have an asymmetry that shifts activity to the right, and another set could shift it to the left. During saccades the activity of these populations would be modulated by eye velocity to produce the desired shifts of stored visual locations.

In Fig. 7, the dashed square inset of the select-all unit's activity is expanded at the bottom of the figure to demonstrate how eye velocity would modulate the remapping response. The activity is shown for both a leftward and rightward selective unit as a leftward saccade brings the target inside the receptive field. The 'predictive' response of the leftward selective unit is enhanced (gray trace). For the rightward unit, it is suppressed (black trace). Quasi-visual cells should exhibit similar modulation if they implement a moving hill type mechanism.

A final prediction for the quasi-visual cells is that they should break into distinct classes with either select-two or select-all selectivity. Tian et al. (2000) investigated the target selectivity of quasi-visual cells but did not test for this distinction. They tested how quasi-visual cells respond when either the next or second to next target is brought inside their receptive field in the memory-guided task. The response was nearly equal regardless of the target. This behavior is consistent with both the select-two or select-all units which respond equally to the next two targets.

Further tests are necessary to resolve the distinction between select-two and select-all type units. The tests performed to date only consider when two targets remain in memory. If tested when three targets remain, we predict that some cells should respond equally to the next two targets, but not the third, while others will respond well to all three. Tests with more targets could further reveal if other selectivities are present (i.e., select-three, select-four, etc.). We consider this in the discussion.

10.3. Build-up units

The select-one units resemble build-up cells (also called visuo-movement cells). They select the location of the next intended target (Schall et al., 1995). The selection occurs through a gradual 'build-up' of activity at the target location prior to the saccade. An example of the build-up is shown on the right in Fig. 7 (second from top). As build-up activity accumulates it inhibits and turns off fixation. At the same time, the select-one units are exciting the burst outputs. This enables burst units to overcome suppression from fixation and initiate a saccade.

We tested the network in a 'no go' paradigm that has been used to study saccade timing and its relation to build-up activity in the FEF (Hanes et al., 1998). Three targets were flashed in parallel in the visual input as shown in Fig. 7. The network selects the brightest of the three for a saccade. The latency of that saccade varies stochastically due to noise in the network units. We implemented a 'no go' condition by activating the stop input 350 ms into the trial. It excites the fixation unit thus suppressing saccades.

The resulting traces from several trials in the 'no go' paradigm are presented in Fig. 8. Traces are shown for a select-one unit at the target location, and also for a select-two unit. Those traces that result in a saccade are shown in black with upward arrows indicating the start of the burst. Those where the saccade was successfully stopped are shown in gray.

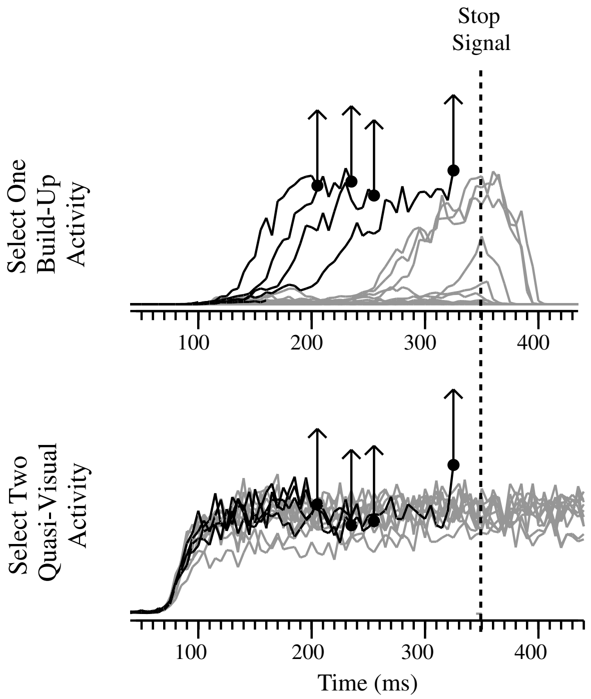

Fig. 8. Traces of unit activity in the go–no go paradigm. Three targets are flashed in the visual input as in Fig. 7. Due to noise, the saccade is initiated at variable latencies. After 350 ms, the stop input is activated to excite fixation which in turn inhibits build-up activity. Several activity traces are shown in each plot for a select-one build-up unit and a select-two quasi-visual unit. For the black traces a saccade is initiated at the time indicated by the upward arrow. Gray traces indicate trials in which no saccade occured.

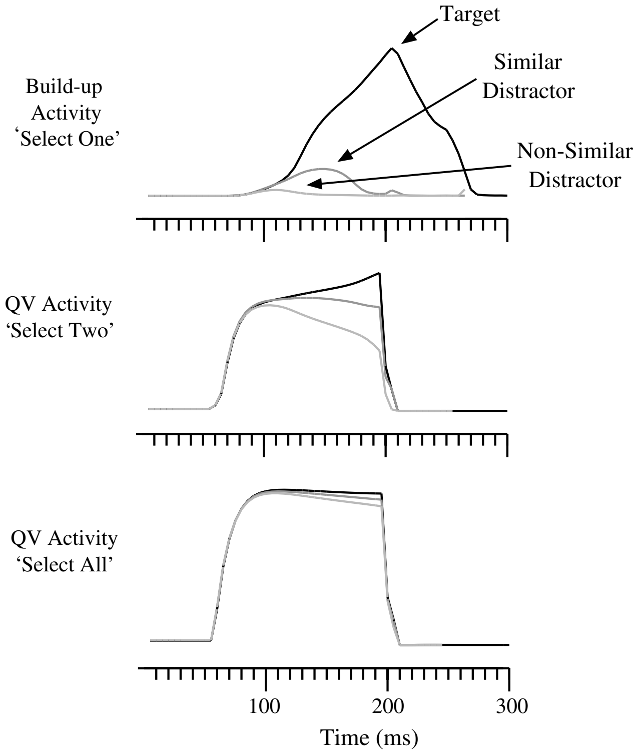

Fig. 9. Partial activation to similar and non-similar distractors. Three potential targets are flashed in a trial similar to Fig. 7. The target is the brightest one, followed by a similar 'distractor' with nearly the same brightness, and a non-similar 'distractor' still less bright (intensities of 1.0, 0.9, and 0.7). The mean response (without noise) at the target location (black), similar distractor location (dark gray), and non-similar distractor location (light gray) is shown for each unit type.

Saccades are initiated when the build-up activity of the select-one unit reaches and persists at a fixed threshold. If it fails to reach that threshold, they do not occur. This closely mimics what is found among build-up cells in the FEF (Hanes & Schall, 1996). In contrast, the select-two unit is not predictive of when saccades occur. This matches what is known for visual tonic cells in the FEF (Murthy, Thompson, & Schall, 2001).

Recent physiological studies also show that build-up cells are partially activated for distractor locations during saccades to the target (Bichot, Rao, & Schall, 2001). Our select-one unit shares this property. In Fig. 9 (top), the mean build-up response is shown when a target, similar distractor, or non-similar distractor are within the unit's receptive field. The magnitude of the partial activation depends on the similarity of the distractor to the target.

Distractor similarity is also reflected in the response of select-two and select-all quasi-visual units. The select-two units respond to both similar and non-similar distractors, but their response to the non-similar distractor is much weaker (Fig. 9, middle). The select-all units do not distinguish distractor similarity in their response (Fig. 9, bottom).

The response to similar and non-similar distractors highlights that the selectivity of units in the network is not as brittle as first presented. The select-one units do not strictly select one target, nor do the select-two units strictly select two. Depending on the similarity of targets, their response can vary, and can show intermediate activation levels. What is important is that the different classes of units have different levels of selectivity. This is what enables them to represent the priority of targets in a distributed fashion.

11. Behavioral comparisons

The behavior of the revised network is compared to humans in several search tasks. It reproduces features of human performance including spatial averaging of close targets, probabilistic selection based on salience, latency vs accuracy trade-offs, inhibition of return, and slower reaction times is search with more distractors.

11.1. Spatial averaging

In double-spot experiments, the network shows spatial averaging of close targets (Ottes, Gisbergen, & Eggermont, 1984). In these tasks two targets of the same

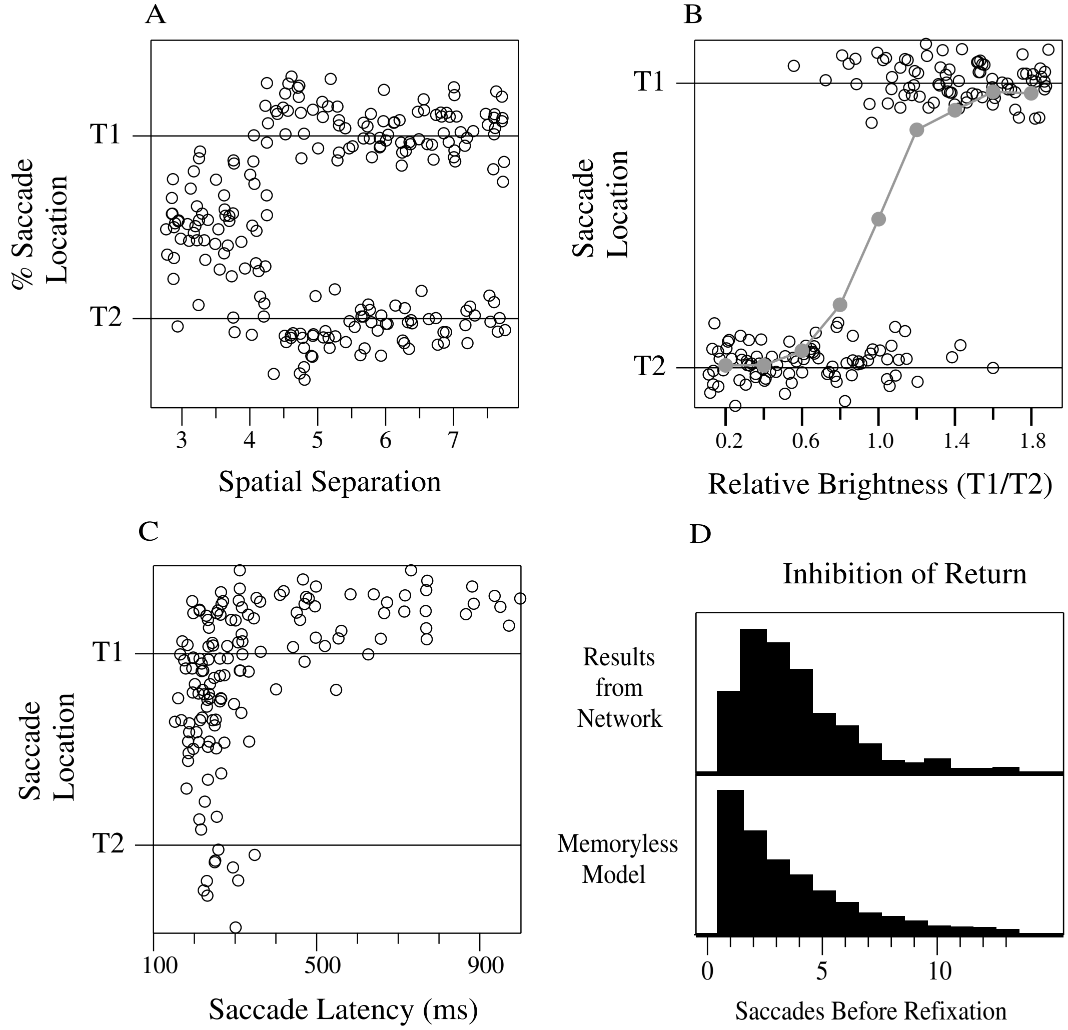

Fig. 10. Behavior in visual search. (A) Two equally bright targets, T1 and T2, are flashed and remain illuminated. The separation between targets is varied (horizontal axis). The distribution of end-points is shown (open circles). The vertical axis gives the end-point location as percentage separation between targets. (B) The double-spot experiment is repeated with the relative brightness of targets T1 and T2 varied. The targets are presented with a spatial separation of 10 units. No spatial averaging occurs, one target or the other is selected probabilistically. The distribution is shown (open circles) with the probability of choice at each value (in gray). (C) The double-spot task is repeated with one target, T1, having a slightly higher color value than the other (I_c of 1.0 and 0.9). The separation of targets is 4 units, a value that normally produces averaging for short latency saccades. Longer latency saccades select the target without averaging. (D) The network is tested for inhibition of return. It is presented with five equally bright targets that remain visible indefinitely. It plans saccades back and forth between them and the order is recorded over 1000 saccades. The number of intervening fixations before each return saccade is calculated and plotted in a histogram (top). The predicted distribution for a memoryless model is exponential (bottom).

intensity are flashed briefly. Humans show two types of behavior depending on the separation between targets. For targets close to each other, the end-points of the first saccade fall somewhere in between the two. For more distant targets one of the two is selected with equal chance.

The distribution of saccade end-points in the double-spot task for different target separations is shown in Fig. 10A. The end-point was determined by the location of peak activity among the burst units. Averaging occurs for target separations within two standard deviations of the Gaussian receptive field size. At this distance the two targets still appear as separate peaks in the visual input. The peak output by burst units falls somewhere between the two visual peaks. This is consistent with that found among real burst cells (Glimcher & Sparks, 1993; van Opstal & van Gisbergen, 1990).

11.2. Probabilistic selection by salience

When two targets are sufficiently separated, the network makes probabilistic choices for one target or the other based on their salience. Several features may contribute to a target's salience. One important feature is its brightness. As the brightness is increased for one target its probability of being chosen increases (Schiller

& Chou, 2000b). In Fig. 10B we plot the distribution of saccades for two targets with different relative intensities. The probability of choosing one over the other is plotted in gray connected dots. The probability increases with target intensity. Other important features in salience are the latency of target onset and duration of its presentation (Schiller & Chou, 2000a). The network prefers targets that appear earlier and for longer durations similar to humans. Varying these features produces similar distributions as seen for intensity in Fig. 10B.

Another feature that influences target selection is the target's proximity to fixation. Targets near fixation are visited before more peripheral ones, even when they are dimmer. A recent study in attention suggests that this bias is due to the magnification of fixation in the cortical representation (Carrasco & Frieder, 1997). Targets that appear near fixation are larger relative to the small foveal receptive fields as compared to the peripheral receptive fields. If the visual input to the network is revised so targets appearing near fixation are larger, then it produces a similar proximal bias.

11.3. Latency and accuracy

We tested if the network exhibited latency and accuracy trade-offs in target selection. Ottes, Gisbergen, and Eggermont (1985) examined this issue using the double-spot paradigm. Two targets of different color but equal intensity were presented with a small spatial separation. Normally, this stimulus produces saccades that are averaged. However, if the targets remain illuminated and subjects are instructed to emphasize accuracy in their response over speed, then they can accurately select a target of a given color. The accuracy of target selection increases with the latency of the saccades.

Revisions of the network's visual input are necessary to perform the same test as Ottes et al. (1985). Previously we have only used phasic visual inputs that encoded intensity differences and turned off after only a few time steps. However, in this task the targets must remain illuminated indefinitely and must be identified by their color.

The response of visual cells to color targets and distractors has been studied in the FEF (Schall et al., 1995; Thompson et al., 1996). Visual cells typically have a phasic and tonic response component. The phasic response does not discriminate a color target from an equi-luminant distractor. However, it typically decays after about 50 ms leaving a tonic response that does discriminate the target. The tonic response is higher when a target matching the desired search color falls inside the receptive field.

We modeled the revised visual input as the sum of a phasic and tonic component. The phasic component encodes the onset of a new stimulus, but not the distinction between a color target and an equi-luminant distractor. The tonic component persists at longer latencies encoding the color differences. Instead of modeling separate color channels, we assume that each stimulus has a single color intensity, I_c, which is higher when it matches the search color. See Section 13 for details.

With the revised input the network produces similar latency and accuracy trade-offs in the color selection task. We presented two targets that had a small spatial separation and remained illuminated indefinitely. One target had a slightly higher color value than the other. The network normally makes fast latency saccades that fall between the two nearby targets (less than 200 ms). In order to compare its performance to humans, we had to simulate the experimental condition in which accuracy is emphasized over speed. To do this, we partially inhibited the select-one units by increasing the value of their negative bias (from -2.8 to -5.8). This slows the rate that activity builds-up to the threshold for saccade initiation thus leading to longer latency saccades. The distribution of endpoints for different latencies is shown in Fig. 10C. Averaging occurs for the shortest latency saccades. At longer latencies the target (T1) is robustly selected.

The increase in accuracy with latency is due to properties of the visual inputs. The sensory information provided becomes more accurate over time as the phasic response decays. Saccades initiated at later delays integrate more accurate information into their decision.

The integration of online visual information is an important feature of the network. Although priorities for targets are set in working memory from the initial visual input, they can still be modified later if the input changes. For example, if the input changes late in a trial to favor a secondary target, that target can push forward to the front of the queue.

The integration of late visual input could be useful to solve tasks in which reflex responses based on pure salience must be replaced with rule-based or cognitive choices. For example, in the anti-saccade task observers must suppress the impulse to make a saccade to a flashed target and instead move in the opposite direction (Fischer & Weber, 1992; Hallett, 1978). Areas representing arbitrary selection criteria could provide a supplementary input to the network. The supplementary input would incur some processing delay, and thus short latency decisions would still be based purely on target salience. However, if the rate of build-up activity was slowed with inhibition, then the supplementary input would have time to influence the decision. At longer saccade latencies, the rule-based target would be selected.

These dynamics are consistent with what is known experimentally in the anti-saccade task. During anti-saccade trials there is an added excitation of fixation cells with a corresponding inhibition of build-up cells

(Everling, Dorris, & Munoz, 1998; Everling & Munoz, 2000). When reflex plans are delayed the initial build-up activity corresponding to the reflex target decays and new activity builds to select the voluntary target.

11.4. Inhibition of return

Next we tested whether the network exhibited inhibition of return in a simple visual search task. Inhibition of return refers to the suppression of previously visited targets during search (Klein, 2000). Debate has surrounded whether such a memory exists, and the extent of its capacity. In some experimental conditions it appears that search is memoryless (Horowitz & Wolfe, 1998). More recent studies suggest that there is some memory, but that it has a small capacity. The probability of return is reduced for the last 3–4 visited targets (Gilchrist & Harvey, 2000).

The network shows a limited capacity inhibition of return. It is tested by presenting five equally bright targets. The targets remain illuminated for an indefinite period and the network plans saccades back and forth between them. The order of targets selected is recorded and the number of saccades before refixation calculated. The distribution of saccades before refixation is plotted in Fig. 10D. If the search is truly memoryless, the chance of making a return is the same at all times. This gives an exponential distribution (bottom plot). The network produces a distribution with a peak around 2–3 saccades (top plot). This indicates a limited inhibition for the last 2–3 targets.

What is the mechanism of inhibition of return? In the network it results not from a memory of past targets, but instead from planning in advance. The targets stored in the queue must be distinct and ordered in sequence. If one target is selected for the current saccade, then it must wait for saccades to subsequent targets to complete before it can return.

The inhibition of return generated here can be distinguished experimentally from an alternative model in which a past memory trace actively suppresses selection (Itti & Koch, 2000). In that model the effect should decay over time along with the memory trace. Our model predicts that it depends on the number of intervening saccades, not the temporal delay. An experiment designed to vary inter-saccadic durations could distinguish these alternatives.

11.5. Reaction times and target number

The reaction time to make a saccade to the target increases with the number of distractors when distractors resemble the target. If the target is sufficiently distinct from the distractors this effect does not necessarily hold, and may even reverse. We test the network with distractors that are similar to the target to determine if its reaction times increase with distractor number. It is tested in trials that include either a single color target, a color target paired with a distractor of a slightly lower color value, or a target paired with two distractors of a slightly lower color value. The distributions of saccadic reaction times for each case are shown in Fig. 11A. Mean reaction time and variability of the first saccade increases with the number of distractors. Similar behavior is seen in humans (Levy-Schoen, 1969; Walker, Deubel, Schneider, & Findlay, 1997). The accuracy of

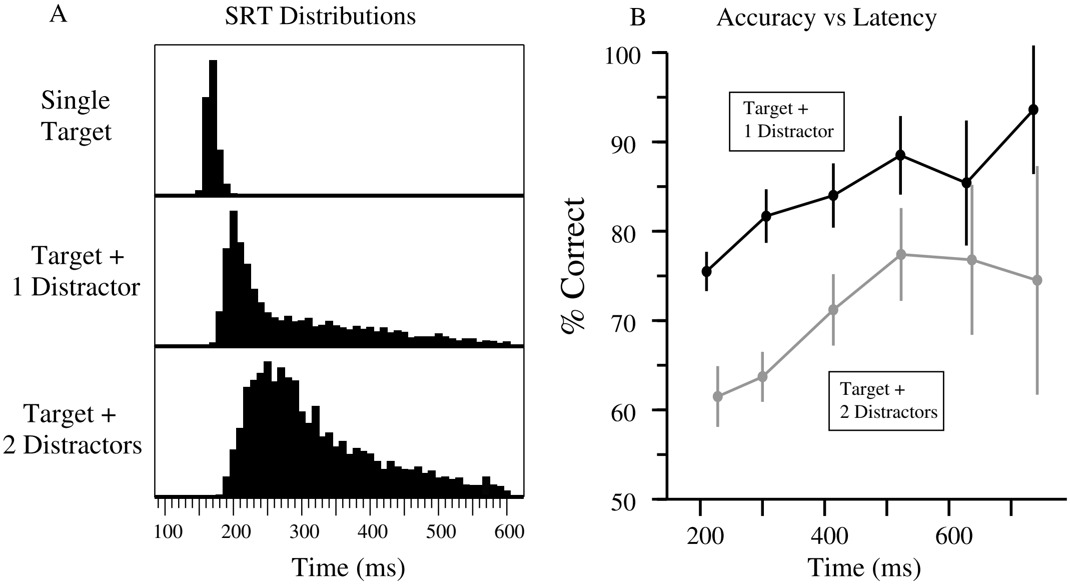

Fig. 11. (A) Saccadic reaction time (SRT) distributions for different numbers of distractors. The color value is 1.0 for the target and 0.80 for distractors. Only trials where the target was correctly selected are included. The saccade onset is measured by when the peak in the output exceeds 0.5. (B) Accuracy of target selection as a function of latency in this task with one distractor (black) and two distractors (gray).

target selection is greater for longer latency saccades (Fig. 11B). This is again due to the properties of the visual inputs. Their tonic response provides more accurate encoding of the color differences that discriminate the target.

We also tested if accuracy would increase as the similarity in color between the target and the distractors was reduced. To model this, we increased the difference between the color values of the distractors and the target. Accuracy improves as the difference is made larger except for the fastest latency saccades which are primarily driven by the non-selective phasic visual input.

The latency of the first saccade of a sequence increases with the length of the sequence. We repeated the experiment in Fig. 11, and allowed the network to make a sequence of saccades that visited first the target and then the distractors. The mean latency of the first saccade increases with the number of targets present, or equivalently, with the number of steps in the sequence. The average was 130 ms for a single step, 218 ms for two steps, and 304 ms for three steps. The lengthening of first saccade latency with sequence length is consistent with human behavior (Sternberg, Monsell, Knoll, & Wright, 1978; Zingale & Kowler, 1987).

The inter-saccadic latency in multi-step saccades reflects that secondary targets are processed concurrently. We measured the latency to initiate saccades to second and third targets. If each saccade were programmed from scratch, then the time necessary to initiate a saccade should be similar regardless of the step. In contrast, the average latency for the first step was 290 ms compared to 118 ms for the second step and 126 ms for the third step. This reflects the advanced planning in the network. Humans behave similarly in multi-step tasks, initiating secondary steps much faster or even as a continuation of the first saccade (Becker & Jurgens, 1979; McPeek & Keller, 2001).

12. Discussion

The memory mechanism identified from training provides a general strategy for prioritizing potential targets and making saccades to them. It works equally well for visual search as for planning memory-guided saccades.

The mechanism is what we call a distributed queue. Units have memory receptive fields for targets, but are not selective to individual targets. Instead each type responds to a different number of targets, selecting one, two, or any. This hierarchy of selectivity is sufficient to encode the same ordered information that is present in a queue. An ordered list of targets can be extracted from it with a simple linear transformation. During saccades the order of targets is updated. The old target is cleared from memory and those remaining shift forward in the hierarchy.

The network units resemble classes of cell found in the FEF during search and sequential saccade tasks. The select-one units resemble build-up or visual movement cells. They respond selectively to the next intended target, accumulating activity immediately prior to a saccade. When their activation reaches a fixed threshold, a saccade is initiated. The latency of saccades can be increased on average by exciting the fixation unit. This produces a uniform inhibition on the select-one units that slows their rate of build-up thus giving longer latency saccades.

The select-two and select-all memory arrays are both consistent with what is currently known about quasi-visual or visual tonic cells. Both classes respond equally well when either the next or second to next target is brought inside their receptive field (Tian et al., 2000). We predict that if tested with more than two targets in memory, the quasi-visual cells previously studied will break into distinct classes, namely that some only respond to the next two targets, while others respond to any.

The mechanism of remapping implemented in our network also makes specific predictions for the behavior of quasi-visual cells. First, they should exhibit a transient burst during saccades if a target passes through their receptive field. Second, the predictive component of their remapping response should be modulated by the direction of the saccade that brings the target into their receptive field.

The network reproduces several features of human performance in search tasks. It exhibits spatial averaging of proximal targets, probabilistic selection based on salience and slower reaction times in visual search with distractors. Previous neural models have shown these effects as well (Clark, 1999; Itti & Koch, 2000; Kopecz & Schoner, 1995). The essential trait that our network shares with these models is its shift-invariant architecture and competitive recurrent interactions.

A new mechanism for inhibition of return is identified. One previous alternative inhibits past target locations for a fixed temporal delay (Itti & Koch, 2000). This prevents return saccades to those locations for a short period. Inhibition of return results in our network because it is planning two to three targets in advance. No repeat targets are included in these plans.

Our presentation has focused on one of the network instantiations identified from training. It was the only one that performed both sequential and parallel tasks perfectly.

A second solution found by a smaller network provides an interesting alternative. This network contained no hidden arrays. It learned to represent the location and order in the array of the memory outputs (the select-all array). It solved both sequential and parallel tasks, but only at the shortest delays. The solution it employs is that of a *spatial map*. Activity is maintained for each target location in a single array. The order of the targets is encoded by graded levels of response at each of their locations. The response is highest for the first target, and progressively weaker for the second and third. At short delays, this is sufficient to program the saccades. At long delays the graded activity decays and weaker targets are forgotten.

It is likely that some mixture of spatial map and distributed queue strategies appear in the FEF. The queue is superior in that it is robust to decay, but it also requires several distinct classes of cells to operate. Each must have a different selectivity to targets. A more efficient strategy for larger numbers of targets may be the spatial map. Although it is not robust to delay, several targets can be encoded by a single type of unit.

The number of targets encoded by a queue could be tested behaviorally. Humans can perform memory-guided sequences with five targets (Ditterich, Eggert, & Straube, 1998). The extent to which this capacity changes for different delays has not been examined systematically. Those targets encoded by a distributed queue should either not decay, or decay with a different time course from those encoded by a spatial map. By testing with multiple targets a different delays it may be possible to distinguish the capacity of the queue.

We have introduced a new strategy for training neural networks in sequential decision tasks. In our strategy, a trial is aborted after an incorrect decision is made. The network is never to trained to make a second or third decision in a sequence unless the preceding decisions are correct.

A related strategy is called *starting small* (Elman, 1994). The networks first learn to perform simple tasks, and then to proceed to more difficult tasks in gradual stages.

Starting small is not well suited to learning the sequential saccade task. Networks can learn a simple task with a single target, but they fail to adapt when a second target is added. Since they have never seen trials with two targets, they do not know which one to choose first, or that ultimately both must be represented. They must unlearn idiosyncrasies acquired from the simple task. The same difficulty arises going from two to three targets.

In our approach the task difficulty remains constant, but training on later decisions is contingent on preceding decisions being correct. This insures that correct state information is represented at each point in the trial. This may be useful for learning other types of sequential task.

13. Methods

13.1. The simulated saccade task

The saccade task is intended to be similar to the triple-step paradigm (Tian et al., 2000). Targets are flashed in sequence. After a delay the fixation cue disappears and saccades are made to each target in the order presented.

The number of targets presented in each trial varies in our task. It can be one, two, or three targets. Trials are randomly interleaved with a 20% chance of a one or a two target trial, and a 60% chance of a three target trial.

The locations of the targets are chosen at random from a continuous range between 0 and 32. The total size of the input array is 32. To insure that two targets do not overlap, a minimum separation of $d = 6$ units is required. Random locations are generated until this criteria is met.

Target intensity is chosen in a semi-random manner. In the sequential task, the intensity is set as a Gaussian random variable with mean of 1.0 and standard deviation of 0.05. In the parallel task, the three targets have intensities selected with uniform chance between 0.7 and 1.3 such that no two targets are closer than 0.15.

The duration of trial events is chosen so their relative sizes are on average similar to that of the triple-step paradigm (Tian et al., 2000). In the triple-step task targets are presented for roughly 60 ms. The duration of saccadic eye movements was similar on average. We choose 3 time steps for the duration of presentations and saccades in our task. Thus one step is comparable to 20 ms.

The timing of events occurs at random. Otherwise, networks can develop solutions that rely on its specifics. The interval between the presentation of targets in a sequence is chosen randomly between 3 and 6 time steps with a uniform chance. The interval between saccades is chosen in the same fashion. The delay between the presentation of the last target and the first saccade is chosen from an exponential distribution with a mean of 3 time steps.

13.2. Input and output encodings

The visual inputs have response properties similar to visual cells in the FEF. They consist of an array of 32 units (Fig. 1 shows only 8 to ease illustration). Each has a light sensitive Gaussian receptive field. Their temporal response has a phasic component that turns off after 3 time steps. The response at array location i to a target with intensity I at retinal location x and appearing at time t_{onset} is given by

$$\text{Phasic}[i] = \begin{cases} I * \mathrm{e}^{-\frac{(i-x)^2}{2\sigma^2}} * \mathrm{e}^{(t_{\text{onset}}-t)/\tau_{\text{V}}} & \text{if } t - t_{\text{onset}} < 3 \\ 0 & \text{else} \end{cases} \tag{7}$$

with $\sigma = 2$ and $\tau_{\text{V}} = 3$ time steps. The response decays exponentially following the stimulus onset and then completely turns-off after three time steps.

A second input to the network is the fixation unit. It maintains a high resting activity of 1.0, and then drops to 0.0 on time steps at which a saccade should be executed.

The saccade outputs are similar to burst cells in the FEF. They also consist of an array of 32 units. They remain silent with zero activity until the time of a saccade. Then a burst occurs at the target location. The desired activation of a burst unit at location i for a saccade to retinal location x is given by

$$\text{Saccade}[i] = \mathrm{e}^{-\frac{(i-x)^2}{2\sigma^2}} \quad (8)$$

with the receptive field size the same as the visual units, $\sigma = 2$. Outputs maintain a constant level of activation throughout the period of the saccade, which lasts for 3 time steps.

The memory outputs respond equally to all of the remaining targets. They consist of an array of 32 units. If there are T targets remaining in the ongoing sequence, then the desired activation of memory output at location i is given as

$$\text{Memory}[i] = \sum_{t}^{T} \mathrm{e}^{-\frac{(i-x_t)^2}{2\sigma^2}} \quad (9)$$

where x_t is the retinal location of each target. The receptive field size is the same as the visual and saccade units, $\sigma = 2$. Their response to a target begins as soon as it appears in visual input and continues until a saccade has been made to it. They maintain a constant level of activity through delay periods.

13.3. Computing synaptic currents by convolutions

The synaptic input currents to units can be rewritten as a sum of convolutions between weight kernels, $\vec{W}$, with arrays of activity, $\vec{Y}$. We use the over right arrow to denote an array of size $N = 32$. The convolution $\vec{S} = \vec{W} * \vec{Y}$ is computed by

$$S[i] = \sum_{n=1}^{N} W[n]Y[i-n] \quad (10)$$

where $S[i]$ is the value at location i in the array $\vec{S}$. Note that the index into $\vec{Y}$, given by $(i-n)$, may be negative or exceed N. To eliminate boundaries, we assume that the arrays wrap around themselves to form a circle in which the last unit is adjacent to the first. A negative index becomes $((i-n)+N)$ and an index greater than N becomes $((i-n)-N)$.

The equations for the net synaptic input currents to the outputs and hidden types are similar. The net current, $\vec{n}_i$, to each array at time $t+1$ is given by

$$\vec{n}_i(t+1) = \sum_{j} \vec{W}_{ij} * \vec{Y}_j(t) + \vec{W}_{iv} * \vec{V}(t) + B_i + F_i * \text{Fix}(t) \quad (11)$$

where $\vec{V}(t)$ is the visual inputs, $\vec{W}_{ij}$ and $\vec{W}_{iv}$ are the weight kernels, B_i is a bias weight, and F_i is the fixation weight and $\text{Fix}(t)$ is the fixation input. The saccade outputs do not receive direct visual input ($\vec{W}_{1v} = 0$) and the memory outputs and hidden units do not receive fixation input ($F_2 = F_3 = F_4 = 0$).

The speed of computing convolutions is much faster if it is done using the FFT (Press et al., 1988). This is possible because the arrays are shift-invariant. The array size must be chosen to be a power of 2. Similar speed ups are also possible for computing the convolutions and correlations that occur in the back-propagation of error during training.

13.4. Training the neural network

The cost function optimized contains three terms. It is given by

$$E = \frac{1}{2}\sum_{k}(Y_1[k]^* - Y_1[k])^2 + \frac{\lambda_1}{2}\sum_{m}(Y_2[m]^* - Y_2[m])^2 + \frac{\lambda_2}{2}\sum_{n} y[n]^2 \quad (12)$$

where Y^* is the desired activity and Y is the activity of the network outputs. The first sum with k indexes the saccade outputs, the second sum with m indexes the memory outputs, and last sum with n indexes all units (both hidden and output). The first term trains the saccade outputs to produce the desired burst activity. The second term trains the memory outputs to maintain a trace of targets until saccades are made to them. It is essential for overcoming long temporal delays in the task. Its relative contribution to the error is set to be small so it does not dominate the saccade outputs which are more difficult to learn ($\lambda_1 = 0.1$). The last term encourages low activity for every unit. It is motivated in that real neurons have low average firing rates. The speed of learning is faster when it is included. It makes a small contribution to the error ($\lambda_2 = 0.001$).

Gradient descent is performed in the space of biases and weight kernel parameters. This requires two modifications to the basic back propagation algorithm. The first is standard for shift-invariant networks. It is called weight sharing. It forces each of the weight patterns in an array to be identical and for each of them to have the same bias value (Fukushima, 1980; LeCun et al., 1989; Rumelhart et al., 1986).

The second modification takes the desired changes to weight kernels (computed from weight sharing) and computes the desired change to the difference of Gaussians parameters (B, A_1, σ_1, A_2, σ_2) for each weight kernel in the network. On each cycle in training, the kernel parameters are updated first, and then the weights are recomputed from them. This constrains gradient descent to the space of DOG parameters. This, along with weight sharing, reduces the complexity of the model.

Weight sharing is implemented on each cycle in training. If the weight to unit i from pre-synaptic input j is given by $w[i][j]$, then $w[i][j] = w[i-s][j-s]$ for every shift s. Thus we only need to store the weights for one of the units. That single set of weights defines what we call the *weight kernel*. The back propagation algorithm computes the changes for weights to every unit as

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \qquad (13)$$

To impose sharing these changes must be averaged across the units for each weight in the kernel. That is

$$\Delta w_j = \frac{\sum_i \Delta w_{i(i+j)}}{N} \qquad (14)$$

where N is the array size. In some cases the index $(i+j)$ may exceed the array size. We assume the array wraps around itself to avoid these boundaries. The weights from fixation and bias inputs are also shared. They are averaged over all units, and a single value stored. All these calculations can be done in the Fourier domain for faster performance.

Gradient descent is performed on the parameters that define the kernels instead of the weights themselves. Using the chain rule, we derive how each DOG parameter, p, should change to reduce the error. It is given as

$$\Delta p = -\eta \frac{\partial E}{\partial p} = -\eta \sum_j \frac{\partial E}{\partial w_j} \frac{\partial w_j}{\partial p} = \sum_j \Delta w_j \frac{\partial w_j}{\partial p} \qquad (15)$$

where j indexes over the weights in the kernel.

Although the update Δp moves in a direction to minimize E, the size of its steps is poorly scaled. This is because the magnitude of the derivatives of the weights with respect to the different DOG parameters vary substantially and thus require a small learning rate to insure stability for all of them. Faster learning is possible if the step sizes are normalized by the magnitude of their derivatives. We use the update $\Delta p' = \frac{1}{Z_p}\Delta p$ where Z_p is the square root of the sum of squared derivatives

$$Z_p^2 = \sum_j \left(\frac{\partial w_j}{\partial p}\right)^2 \qquad (16)$$

At the start of training, the parameters for the weight kernels are initialized randomly in a region close to winner-takes-all behavior. This is done first by setting the bias parameter to a small negative value, $B = -0.1$, and the width parameters, σ_1 and σ_2, as 1.0 and 2.5. Then the amplitude parameters are chosen randomly. For the weight kernel that connects arrays to themselves and to visual input (W_{ii} and W_{iv}) the excitatory amplitude is chosen between 1 and 4 and the inhibitory amplitude between −0.5 and −1.5. This produces winner-takes-all dynamics on average. For other kernels amplitudes are initialized randomly to small random values between −0.2 and 0.2.

13.5. Revisions of the fixation signal

The fixation signal was implemented as a single neuron-like unit with connections to the burst and select-one arrays. The synaptic input, net_f, at time $t+1$ was given as

$$\text{net}_\text{f}(t+1) = x_\text{stop}(t) + 2.5x_\text{f}(t) - 6\overline{Y_0}(t) - 18\overline{Y_1}(t) \qquad (17)$$

where x_stop is the value of the stop input activity at time t, x_f is the value of fixation activity, $\overline{Y_0}$ is the average activity of the array of burst outputs, and $\overline{Y_1}$ is the average activity of the array of select-one units. The fixation activation on the next time step, $x_\text{f}(t+1)$, is computed from the synaptic input as described previously for the other network units using a continuous time integration followed by a sigmoid shaped nonlinearity.

The fixation unit inhibits each of the units in the burst and select-one arrays. It connects to each of the burst units with a strong inhibitory weight of −12, and to each of the select-one units with a weight of −4. In this way, fixation competes with burst and select-one arrays for activity.

13.6. The 'stop' input and target clearing

The remapping during saccades was terminated by turning on the stop input. The activity in the stop input rose sharply as the end of the saccade was approached. During the saccade, we tracked the active location corresponding to the target in the select-all array of units. The stop input activity was then given as

$$x_\text{stop} = \begin{cases} 3.1 \ * \ (8-D) & \text{if } D < 8 \\ 0 & \text{else} \end{cases} \qquad (18)$$

where D was the distance (in array units) of the target's active location from the fixation zone. The excitation of fixation from the stop input ultimately turned off the saccade. Once the saccade ended, the stop input was reset to zero. A saccade was considered 'turned off' if the maximum activation in the array of burst outputs fell below a value 0.5 (conversely a saccade was considered 'turned on' when the maximum activation exceeded that value). In the memory task the stop input was held on to a value of 4.0 to prevent saccade initiation during the delay period.

Clearing at the fixation zone was implemented by adding a negative bias to the units at fixation. A Gaussian kernel with spatial width $\sigma = 2.0$ and negative magnitude of −2 and −1 was added respectively to the bias of units at the centers of the select-all and select-two arrays. Activity at the center thus decays clearing fixated targets from memory.

13.7. Revised visual inputs

We modeled the revised visual input as the sum of a phasic and tonic component for color discrimination tasks. The phasic component encodes the onset of a new stimulus, but not the distinction between a colored target and an equi-luminant distractor. The tonic component which dominates at longer latencies does encode the color differences. Instead of modeling separate color channels, we assume each stimulus has a color intensity, I_c, which is higher for a stimulus that matches the desired search color. These details are consistent with the selection behavior of visual cells in FEF during search tasks (Schall et al., 1995; Thompson et al., 1996).

The phasic component of the response resembles a magnocellular response in early visual processing. It is less accurate spatially and does not encode the color or form of stimuli. The activation at array location i is given as

$$\text{Phasic}[i] = e^{-\frac{(i-x)^2}{2\sigma_p^2}} * e^{(t_{\text{onset}}-t)/\tau_v} \qquad (19)$$

with $\sigma_p = 4$, $\tau_v = 60$ ms, and t_{onset} is the time of stimulus onset. In modeling reaction time data we assume an additional delay of 67 ms to the value of t_{onset} in order to model the time it takes a stimulus presented on the retina to reach the FEF visual cells. This value is consistent with typical visual latencies of FEF cells Thompson et al. (1996).

The tonic component of the visual response is more accurate spatially and encodes color differences between the target and the distractors. The activation for a target of color intensity I_c at array location i is

$$\text{Tonic}[i] = e^{-\frac{(i-x)^2}{2\sigma_t^2}} * I_c * 0.2 * (1 - e^{(t_{\text{onset}}-t)/\tau_v}) \qquad (20)$$

with I_c the color intensity, t_{onset} and τ_v defined the same as above, and $\sigma_t = 2$. This response grows from the stimulus onset until it reaches a constant value that is about 20% the magnitude of the initial phasic response.

Appendix A. Revised network parameters

The code for a Matlab demo is available at www.snl.salk.edu/~jude.

$N = 64$ (number of units in each array)
$\Delta t = 1$ ms
$B_1 = B_2 = B_3 = B_4 = -2.8$
$A_2 = 0.2 * A_1$ for all kernels
$\sigma_1 = 0.88$ and $\sigma_2 = 2.03$ for all kernels

Kernel	B	A_1
$\vec{W}_{2v}$	−0.05	+2.00
$\vec{W}_{3v}$	−0.05	+2.00
$\vec{W}_{4v}$	−0.05	+1.00
$\vec{W}_{11}$	−1.50	+15.0
$\vec{W}_{12}$		
$\vec{W}_{13}$		
$\vec{W}_{14}$	−0.60	+4.0
$\vec{W}_{21}$		
$\vec{W}_{22}$	−0.10	+6.0
$\vec{W}_{23}$		
$\vec{W}_{24}$		
$\vec{W}_{31}$		
$\vec{W}_{32}$	−0.10	+2.0
$\vec{W}_{33}$	−0.40	+6.0
$\vec{W}_{34}$		
$\vec{W}_{41}$		
$\vec{W}_{42}$		
$\vec{W}_{43}$	−0.10	+3.0
$\vec{W}_{44}$	−0.60	+8.0

References

Anderson, R., Keller, E., Gandhi, J., & Das, S. (1998). Two-dimensional saccade-related population activity in superior colliculus in monkey. *Journal of Neurophysiology, 80*, 798–817.

Becker, W., & Jurgens, R. (1979). Analysis of the saccadic system by means of double-step stimuli. *Vision Research, 19*, 967–983.

Bichot, N., Rao, S. C., & Schall, J. (2001). Continuous processing in macaque frontal cortex during visual search. *Neuropsychologia, 39*, 972–982.

Carrasco, M., & Frieder, K. (1997). Cortical magnification neutralizes the eccentricity effect in visual search. *Vision Research, 37*(1), 63–82.

Chafee, M., & Goldman-Rakic, P. (1998). Matching patterns of activity in primate prefrontal area 8a and parietal area 7ip neurons during a spatial working memory task. *Journal of Neurophysiology, 79*, 2919–2940.

Clark, J. (1999). Spatial attention and latencies of saccadic eye movements. *Vision Research, 39*(3), 585–602.

Ditterich, J., Eggert, T., & Straube, A. (1998). Fixation errors and timing in sequences of memory-guided saccades. *Behavioural Brain Research, 95*, 205–217.

Droulez, J., & Berthoz, A. (1991). A neural network model of sensoritopic maps with predictive short-term memory properties. *Proceedings of the National Academy of Science, 88*, 9653–9657.

Elman, J. (1994). Implicit learning in neural networks: the importance of starting small. In *Attention and performance 15: conscious and nonconscious information processing* (Vol. 15, pp. 861–888). Cambridge, MA: MIT Press.

Everling, S., Dorris, M., Klein, R., & Munoz, D. (1999). Role of primate superior colliculus in preparation and execution of anti-saccades and pro-saccades. *Journal of Neuroscience, 19*, 2740–2754.

Everling, S., Dorris, M., & Munoz, D. (1998). Reflex suppression in the anti-saccade task is dependent on prestimulus neural processes. *Journal of Neurophysiology, 80*, 1584–1589.

Everling, S., & Munoz, D. (2000). Neural correlates of preparatory set associate with pro-saccades and anti-saccades in the primate frontal eye fields. *Journal of Neuroscience, 20*, 387–400.

Everling, S., Pare, M., Dorris, M., & Munoz, D. (1998). Comparison of the discharge characteristic of brain stem omnipause neurons and superior colliculus fixation neurons in monkey: implications for control of fixation and saccade behavior. *Journal of Neurophysiology, 79*, 511–528.

Fischer, B., & Weber, H. (1992). Characteristics of "anti" saccades in man. *Experimental Brain Research, 89*, 415–424.

Fukai, T., & Tanaka, S. (1997). A simple neural network exhibiting selective activation of neuronal ensembles: from winner-take-all to winners-share-all. *Neural Computation, 9*(1), 77–97.

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics, 36*, 193–202.

Gilchrist, I., & Harvey, M. (2000). Refixation frequency and memory mechanisms in visual search. *Current Biology, 10*, 1209–1212.

Glimcher, P., & Sparks, D. (1993). Representation of averaging saccades in the superior colliculus of the monkey. *Experimental Brain Research, 95*, 429–435.

Goldberg, M., & Bruce, C. (1990). Primate frontal eye fields. iii. Maintenance of a spatially accurate saccade signal. *Journal of Neurophysiology, 64*, 489–508.

Grossberg, S., Roberts, K., Aguilar, M., & Bullock, D. (1997). A neural model of multimodal adaptive saccadic eye movement control by superior colliculus. *Journal of Neuroscience, 17*, 9706–9725.

Hahnloser, R., Douglas, R., Mahowald, M., & Hepp, K. (1994). Feedback interactions between neuronal pointers and maps for attentional processing. *Nature Neuroscience, 2*(8), 746–752.

Hallett, P. (1978). Primary and secondary saccades to goals defined by instructions. *Vision Research, 18*, 1279–1296.

Hanes, D., Patterson, W., & Schall, J. (1998). Role of frontal eye fields in countermanding saccades: visual, movement, and fixation activity. *Journal of Neurophysiology, 79*, 817–834.

Hanes, D. P., & Schall, J. D. (1996). Neural control of voluntary movement initiation. *Science, 274*, 427–430.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward network are universal approximators. *Neural Networks, 2*, 359–366.

Horowitz, T., & Wolfe, J. (1998). Visual search has no memory. *Nature, 394*, 575–577.

Isoda, M., & Tanji, J. (2002). Cellular activity in the supplementary eye field during sequential performance of multiple saccades. *Journal of Neurophysiology, 88*, 3541–3545.

Itti, L., & Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research, 40*(10–12), 1489–1506.

Klein, R. (2000). Inhibition of return. *Trends in Cognitive Sciences, 4*, 138–147.

Kopecz, K., & Schoner, G. (1995). Saccadic motor planning by integrating visual information and pre-information on neural dynamic fields. *Biological Cybernetics, 73*, 49–60.

Krommenhoek, K., Opstal, A. V., Gielen, C., & Gisbergen, J. V. (1993). Remapping of neural activity in the motor colliculs: a neural network study. *Vision Research, 33*(9), 1287–1298.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*, 541–551.

Levy-Schoen, A. (1969). Determination et latence de la reponse oculomotrice a deux stimulus. *L'Annee Psychologique, 69*, 373–392.

Lu, X., Matsuzawa, M., & Hikosaka, O. (2002). A neural correlate of oculomotor sequences in supplementary eye field. *Neuron, 34*, 317–325.

McPeek, R., & Keller, E. (2001). Superior colliculus activity related to concurrent processing of saccade goals in a visual search task. *Journal of Neurophysiology, 87*, 1805–1815.

Mitchell, J., & Zipser, D. (2001). A model of visual–spatial memory across saccades. *Vision Research, 41*, 1575–1592.

Moody, S., Wise, S., Pellegrino, G., & Zipser, D. (1998). A model that accounts for activity in primate frontal cortex during a delayed matching-to-sample task. *Journal of Neuroscience, 18*(1), 399–410.

Munoz, D., & Guitton, D. (1991). Control of orienting gaze shifts by the tectoreticulospinal system in the head-free cat. ii. Sustained discharges during motor preparation and fixation. *Journal of Neurophysiology, 66*, 1624–1641.

Munoz, D., Guitton, D., & Pelisson, D. (1991). Control of orienting gaze shifts by the tectoreticulospinal system in the head-free cat. iii. Spatiotemporal characteristics of phasic motor discharges. *Journal of Neurophysiology, 66*, 1642–1666.

Munoz, D., Waitzman, D., & Wurtz, R. (1996). Activity of neurons in monkey superior colliculus during interrupted saccades. *Journal of Neurophysiology, 75*, 2562–2580.

Munoz, D., & Wurtz, R. (1993a). Fixation cells in the monkey superior colliculus. i. Characteristics of cell discharge. *Journal of Neurophysiology, 70*, 559–575.

Munoz, D., & Wurtz, R. (1993b). Fixation cells in the monkey superior colliculus. ii. Reversible activation and deactivation. *Journal of Neurophysiology, 70*, 576–589.

Munoz, D. P., & Istvan, P. J. (1998). Lateral inhibitory interactions in the intermediate layers of the monkey superior colliculus. *Journal of Neurophysiology, 79*, 1193–1209.

Murthy, A., Thompson, K., & Schall, J. (2001). Dynamic dissociation of visual selection from saccade programming in frontal eye field. *Journal of Neurophysiology, 86*, 2634–2637.

Ottes, F., Gisbergen, J. V., & Eggermont, J. (1984). Metric of saccade responses to visual double stimuli: two different modes. *Vision Research, 24*(10), 1169–1179.

Ottes, F., Gisbergen, J. V., & Eggermont, J. (1985). Latency dependence of colour-based target vs nontarget discrimination by the saccade system. *Vision Research, 25*, 849–862.

Pare, M., & Wurtz, R. (2001). Progression in neuronal processing for saccadic eye movements from parietal cortex area lip to superior colliculus. *Journal of Neurophysiology, 85*(6), 2545–2562.

Pearlmutter, B. (1995). Gradient calculations for dynamic recurrent neural networks: a survey. *IEEE Transactions on Neural Networks, 6*(5), 1212–1228.

Pierrot-Deseilligny, C., Rivaud, S., Gaymard, B., Muri, R., & Vermersch, A. (1995). Cortical control of saccades. *Neurological Progress, 37*, 557–567.

Press, W., Vetterling, W., Teukolsky, S., & Flannery, B. (1988). *Numerical recipes in C: The art of scientific computing*. New York, NY: Cambridge University Press.

Pylyshyn, Z., & Storm, R. (1988). Tracking multiple independent targets: evidence for a parallel tracking mechanism. *Spatial Vision, 3*, 179–197.

Reynolds, J., Pasternak, T., & Desimone, R. (2000). Attention increases sensitivity of v4 neurons. *Neuron, 26*, 703–714.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propogation. In *Parrallel distributed processing: Explorations in the microstructures of cognition* (Vol. 1, pp. 316–362). Cambridge, MA: MIT Press.

Schall, J. D., Hanes, D. P., Thompson, K. G., & King, D. J. (1995). Saccade target selection in frontal eye field of macaque, i. Visual and premovement activation. *Journal of Neuroscience, 15*(10), 6905–6918.

Schiller, P., & Chou, L. (2000a). The effects of anterior arcuate and dorsomedial frontal cortex lesions on visually guided eye movements in the rhesus monkey: 1. Single and sequential targets. *Vision Research, 40*(10–12), 1609–1626.

Schiller, P., & Chou, I. (2000b). The effects of anterior arcuate and dorsomedial frontal cortex lesions on visually guided eye movements in the rhesus monkey: 2. Paired and multiple targets. *Vision Research, 40*(10–12), 1627–1638.

Schlag, J., Dassonville, P., & Schlag-Rey, M. (1998). Interaction of the two frontal eye fields before saccade onset. *Journal of Neurophysiology, 79*(1), 64–72.

Scudder, C., Kaneko, C., & Fuchs, A. (2002). The brainstem burst generator for saccadic eye movements. *Experimental Brain Research, 142*, 439–462.

Sears, C., & Pylyshyn, Z. (2000). Multiple object tracking and attentional processing. *Canadian Journal of Experimental Psychology, 54*, 1–14.

Segraves, M. (1992). Activity of monkey frontal eye field neurons projecting to oculomotor regions of the pons. *Journal of Neurophysiology, 68*, 1967–1985.

Segraves, M., & Goldberg, M. (1987). Functional properties of corticotectal neurons in the monkey's frontal eye field. *Journal of Neurophysiology, 58*, 1387–1419.

Shima, K., & Tanji, J. (2000). Neuronal activity in the supplementary and presupplementary motor areas for temporal organization of multiple movements. *Journal of Neurophysiology, 84*, 2148–2160.

Soetedjo, R., Kaneko, C., & Fuchs, A. (2001). Evidence against a moving hill in the superior colliculus during saccadic eye movements in the monkey. *Journal of Neurophysiology, 87*, 2778–2789.

Softky, W., & Koch, C. (1996). The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps. *Journal of Neuroscience, 13*(1), 334–350.

Sommer, M., & Tehovnik, E. (1999). Reversible inactivation of macaque dorsomedial frontal cortex: effects on saccades and fixations. *Experimental Brain Research, 124*, 429–446.

Sommer, M., & Wurtz, R. (2001). Frontal eye fields send delay activity related to movement, memory, and vision to the superior colliculus. *Journal of Neurophysiology, 85*(4), 1673–1685.

Sparks, D. (1989). The neural encoding of the location of targets for saccadic eye movements. *Journal of Experimental Biology, 146*, 195–207.

Sternberg, S., Monsell, S., Knoll, R., & Wright, C. (1978). The latency and duration of rapid movement sequences: comparisons of speech and type writing. In *Information processing in motor control*. New York, NY: Academic Press.

Tanji, J., & Shima, K. (1994). Role for supplementary motor area cells in planning several movements ahead. *Nature, 371*, 413–416.

Thompson, K., Hanes, D., Bichot, N., & Schall, J. (1996). Perceptual and motor processing stages identified in the activity of macaque frontal eye field neurons during visual search. *Journal of Neurophysiology, 76*(6), 4040–4055.

Tian, J., Schlag, J., & Schlag-Rey, M. (2000). Testing quasi-visual neurons in the monkey's frontal eye field with the triple-step paradigm. *Experimental Brain Research, 130*(4), 433–440.

Trappenberg, T., Dorris, M., Munoz, D., & Klein, R. (2001). A model of saccade initiation based on the competitive integration of exogenous and endogenous signals in the superior colliculus. *Journal of Cognitive Neuroscience, 13*(2), 256–271.

Umeno, M., & Goldberg, M. (1997). Spatial processing in the monkey frontal eye field, i. Predictive visual responses. *Journal of Neurophysiology, 78*, 1373–1383.

Umeno, M., & Goldberg, M. (2001). Spatial processing in the monkey frontal eye field, ii. Memory responses. *Journal of Neurophysiology, 86*, 2344–2352.

van Opstal, A., & van Gisbergen, J. (1990). Role of monkey superior colliculus in saccade averaging. *Experimental Brain Research, 79*, 143–149.

Walker, R., Deubel, H., Schneider, W., & Findlay, J. (1997). Effect of remote distractors on saccade programming: evidence for an extended fixation zone. *Journal of Neurophysiology, 78*(2), 1108–1119.

Wurtz, R., & Optican, L. (1994). Superior colliculus cell types and models of saccade generation. *Current Opinion in Neurobiology, 4*, 857–861.

Wurtz, R., Sommer, M., Pare, M., & Ferraina, S. (2001). Signal transformations from cerebral cortex to superior colliculus for the generation of saccades. *Vision Research, 41*(25–26), 3399–3412.

Xing, J., & Andersen, R. (2000). Memory activity of lip neurons for sequential eye movements simulated with neural networks. *Journal of Neurophysiology, 84*, 651–665.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience, 16*(6), 2112–2126.

Zingale, C., & Kowler, E. (1987). Planning sequences of saccades. *Vision Research, 27*, 1327–1341.

Zipser, D. (1992). Identification models of the nervous system. *Neuroscience, 47*, 853–862.