To appear in: *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, Chicago 1991

Efficient Visual Search: A Connectionist Solution

Subutai Ahmad and Stephen Omohundro

International Computer Science Institute,
1947 Center Street, Suite 600, Berkeley, CA 94704.
ahmad@icsi.berkeley.edu, om@icsi.berkeley.edu

Abstract

Searching for objects in scenes is a natural task for people and has been extensively studied by psychologists. In this paper we examine this task from a connectionist perspective. Computational complexity arguments suggest that parallel feed-forward networks cannot perform this task efficiently. One difficulty is that, in order to distinguish the target from distractors, a combination of features must be associated with a single object. Often called the *binding problem*, this requirement presents a serious hurdle for connectionist models of visual processing when multiple objects are present. Psychophysical experiments suggest that people use covert visual attention to get around this problem. In this paper we describe a psychologically plausible system which uses a focus of attention mechanism to locate target objects. A strategy that combines top-down and bottom-up information is used to minimize search time. The behavior of the resulting system matches the reaction time behavior of people in several interesting tasks.

Introduction

In 1986, Sejnowski wrote: "The binding problem is a touchstone for testing network models that claim to have psychological validity" (Sejnowski 1986). In 1991, the statement is still true. In visual search two aspects of the problem are important: feature integration and localization. Feature integration is concerned with the interference between features of different objects when a parallel representation is used. Consider an image with red, blue, vertical and horizontal objects. By computing a global OR of appropriate feature maps, one can detect in parallel which colors and orientations are present in the image. However, to detect a red and horizontal object in the presence of other objects one would have to pre-compute every possible conjunction of features at every location. Similarly, the interference between objects makes it difficult to recover the locations of individual objects. With a selective attention mechanism that inihibits all but the features of a single object, the interference is removed and the binding problem goes away. Such a model implies that in some situations a serial search is required. This line of reasoning is used to explain experimental results on visual search (reviewed in (Treisman 1988)). The original experiments showed that search for targets defined by a single feature can be computed in parallel but that targets defined by a conjunction of two features required time linear in the total number of objects. (See, however, the section on simulations for exceptions to this rule.)

For a network implementation of visual search to be useful as well as psychologically plausible, a number of constraints must be met. The system should work for high resolution images and must therefore be efficient along several dimensions. The complexity of the network should be low. The time per attention shift should be small. Covert attention shifts in people take about 40-60msecs (Julesz & Bergen 1987). Since neurons can only fire every 5-10 msecs, this leaves time for at most 8-12 sequential steps. The system must be able to deal with objects that vary continuously in size. Finally, the serial component of the search process should be as small as possible, so the number of successive fixations should be minimized.

We have previously described an efficient mechanism for selective attention in the context of a connectionist network for computing spatial relations (Ahmad & Omohundro 1990a). In the following section we describe an extended version for modeling visual search. This network meets the efficiency constraints listed above. A parallel search strategy, SWIFT, is used to minimize search time. In a final section we present simulation results of the system and discuss its relation to recent experimental results on visual search.

A Model Of Visual Attention

In this section we describe a connectionist model of covert visual attention (see Figure 1). A set of basic features are first computed from the image. The information is then fed to two different systems: a gating network and a priority network. The gating network implements the focus - its function is to restrict higher level processing to a single circular region. The priority network ranks image locations in parallel according to their relevance to the current task. Finally, a set of control networks are responsible for mediating the information flow between these two networks, as well as incorporating top-down knowledge. Each of these parts are described in more detail below.

The Feature Maps

Feature maps in the network are analogous to the topographic maps early in the visual system. A set of basic features (orientation, color, etc.) are detected at each pixel in

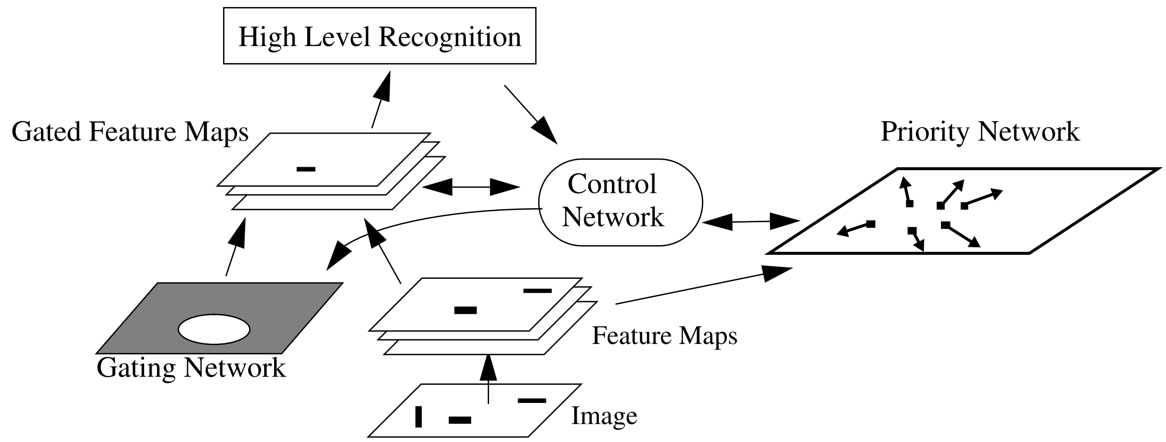

Figure 1 . An overview of the network.

the image in parallel, using one unit at each location for every feature. In addition there is a unit for each feature map which computes the global sum of the activity in the map. Exactly which features should be included is an active area of research. For our purposes, any local feature may be used. Our current implementation uses four feature maps: red, blue, horizontal, and vertical.

The Gating Network and Gated Feature Maps

To tackle the binding problem the network must be able to inhibit the transmission of features to the recognition stage. This is accomplished by the gating network and the gated feature maps. The gating network contains one unit per pixel. Each gate unit receives as input three parameters (A_x,A_y, and A_r) representing the center and radius of the current circular focus of attention. Only the gate units *outside* the circle turn on (see (Ahmad and Omohundro 1990a) for details). Each unit within the gated feature maps receives activation from the corresponding feature detector and inhibition from a gate unit (Figure 1). Only the portion of the gated feature map that falls within the current focus will respond. The resulting system can filter image properties based on an external control signal. As with regular feature maps, the network computes a global sum for each gated feature map. When attention is focused on an object, the activity of these sum units will reflect the object's features, regardless of distractors. Retrieving the location of an object is simple: with attention centered on the object, the units representing A_x, and A_y will reflect the object's location. The complexity of the network is linear in the number of pixels and the time to focus is a small constant.

An alternate architecture would use direct inhibition of the feature maps themselves. This would eliminate the need for a separate set of gated maps. However, in a focused state, such a network would be unable to make global decisions based on the features. With the configuration described above, the network can efficiently access both local and global information simultaneously. As we will see below, this ability is crucial in efficiently carrying out visual search. There is even some direct psychological evidence to support the current architecture. When attention is highly focused, people are able to report primitive features of stimuli appearing outside the focus of attention (Rock et al. 1990) but they are unable to report shape information suggesting that higher level processing is affected.

The Priority Network

This gating method relies on an external mechanism for determining focus locations and this is provided by the priority network. Its job is to rank image locations in order of importance and to help shift the focus to those locations. The main component is a coarse coded map in which the output of each unit reflects the priority of the region within its receptive field. A simple and efficient way to rank regions is to use the mass of the points within the receptive field (Ahmad & Omohundro 1990a). If this is the only ranking available, however, then search for attributes other than size could be inefficient. Several psychophysical experiments have pointed out other possibilities. (Yantis & Jonides 1990) provide evidence that stimuli which appear abruptly are attended to sooner than persistent stimuli, but that this can be overriden by explicit instructions to the subject. Experiments on visual search suggest that objects with the same features or form as the target object can get higher priority than other objects (Egeth, Virzi, & Garbart 1984). All of this suggests a much more dynamic and flexible priority system than one which simply ranks the locations based on pixel density.

This sort of flexibility can be added to the network without sacrificing efficiency. The priority network contains a unit associated with each feature map, P_f, whose value indicates the importance of that map. This value is dynamically adjusted according to the task. The priority of units in the error map is computed as:

$$A_i = G\left(\sum_{x,\, y\, \in\, RF_i}\sum_{f \in F} P_f A_{fxy}\right)$$

A_{fxy} is the activation of the feature unit at location *(x,y)*. RF_i denotes the receptive field of unit *i,* and *G* is a monotonically increasing function of its input (we use a sigmoid). When P_f is *0*, feature *f* has no effect on the priority map. This allows the system to completely shut off the effect of any feature map in parallel.

We also need a way to update the focus of attention to the relevant locations. To accomplish this each location in the priority map contains two additional units whose values encode an "error vector". The error vector is simply the difference between the units' location and the current center of focus. These vectors are constantly updated as the focus moves around. This representation is flexible and efficient. To move the focus to the highest priority location, the control network simply chooses the corresponding error vector and adds its components to A_x and A_y. To choose the nearest location, the control network selects the smallest error vector. To choose locations to the right it can select a vector whose first component is positive.[1]

The Control Network

The control network coordinates the information flow between the gating network and the priority network. It consists of a collection of autonomous sub-networks carrying out independent tasks. There are networks which continually fine tune the scale and location of the focus of attention, networks for storing locations, and a network for updating the focus to the next error vector. These are described in detail in (Ahmad & Omohundro 1990ab). The main addition is the subsystem SWIFT which controls the search process. This is described below.

The SWIFT search strategy. The main function of SWIFT is to integrate top-down and bottom-up knowledge to efficiently guide the search process. Top down information about the target features are stored in a set of units. Let *T* be this set of features. Since the desired object must contain *all* the features in T, any of the corresponding feature maps may be searched. Using the ability to weight feature maps differently, the SWIFT network can remove the influence of all but one of the features in *T.* By setting this map's priority to 1, and all others to 0, the system will effectively prune objects which do not contain this feature. (Hence the name SWIFT: Search WIth Features Thrown out.) To minimize search time, it should choose the one corresponding to the least number of objects. Since it is difficult to count the number of objects in parallel, the network chooses the map with the minimal total activity as the one likely to contain the minimal number of objects.

SWIFT was inspired by the experiments in (Egeth, Virzi, & Garbart 1984). These authors present evidence suggesting that serial search can be restricted to objects with a particular feature. For example, if subjects were instructed to attend to red objects, and the number of red objects were kept small and constant, then search time was constant with respect to the number of distractors. In our implementation, the system dynamically computes the best feature.

Simulations

The simulation proceeds as follows. Initially the network is presented with an image and "shown" the target object by focusing attention on it. The network stores the activity of the gated feature maps in a set of units and these become the target features. For each subsequent image, the total activity of all the feature maps is computed in parallel. Among the target features, the network chooses the one with the least activity and sets its priority to *1* and all others to *0*. Search then proceeds by sequentially visiting locations in order of their saliency. As the focus of attention stabilizes on each location, the control network checks the features of the current object against the stored target representation. This continues until a match is found or there are no more objects.

The output of the simulator is shown in Figure 3(a). The lower left quadrant displays the image (the one shown is 64x64 pixels). The top left quadrant displays the activity of the four gated feature maps. Clockwise from top left the features are: blue, red, vertical and horizontal. In the figure the system is attending to the leftmost red-vertical object, so only the units at that location are active. The bottom right quadrant shows the feature maps that are currently affecting the priority map. Since the target is a blue-vertical object, the system has chosen the vertical map as the minimal feature map. The top right quadrant displays the error vectors in the priority map. Note that only vertical objects have significant priority. The run-time behavior of the network is discussed in the following sections.

Search Time With SWIFT

Since SWIFT always searches the minimal feature map, the critical variable, *M*, that determines search time is:

$$M = \min_{f \in T} \{ O(f) \}$$

where *f* ranges over all the target object's features, and *O(f)* is the number of objects with feature *f*. Search time will always be linear in *M*, but does not necessarily have anything to do with *D*, the number of distractors. For example, in images such as in Figure 3(b), the vertical map will be chosen as the minimal feature map. Search time will not depend on the number of horizontal items. In a sense the search time is dependent on the discriminability of the target object and not on the total number of distractors. Figure 4 plots the actual search time averaged over several trials for various combinations of *M* and *D*. In Figure 4(a), the number of distractors is fixed at *40* as *M* is gradually increased. As expected, mean search time in-

1. The error vector representation was inspired by a similar mechanism for controlling eye saccades in the monkey superior colliculus (Sparks 1986).

creases linearly. Since the search is self-terminating, the ratio of the slopes for the target absent and target present cases is about 2:1. In Figure 4(b), the graphs show that search time can remain relatively flat as D increases, as long as M is held constant. To our knowledge this specific set of experiments has not been performed on people. In the following sections we discuss SWIFT in relation to many of the experiments that have been done.

Relationship With Psychological Data

Single and conjunctive feature searches. We first show that the original search results (Treisman 1988) can be replicated with SWIFT. The experiments showed that targets defined by a single feature (e.g. a red target among blue objects) can be detected in parallel. Targets defined by a conjunction of two features (e.g. a red-horizontal target among red-vertical and blue-horizontal objects) required time linear in the number of objects.

For single feature searches, T contains one feature so SWIFT will always choose it. To detect whether the target is present just requires one step since there can be at most one object with that feature. For conjunction searches T contains two features. If the number of objects with each feature is chosen randomly, on average M will be $1/2D$. Therefore average search time will grow linearly with D (see (Egeth, Virzi, & Garbart 1984) for a similar argument). The ratio of slopes for images with target absent to target present will be *2:1*, consistent with any self-terminating serial scan. More recently it was shown that accurately detecting conjunctions depended on accurate localization of the target (Treisman & Sato 1990). This is also consistent with our architecture (Ahmad & Omohundro, 1990a).

Triple conjunction search. Search for an object defined by a conjunction of three features results in different search slopes (Quinlan & Humphreys 1987). There were two situations that were tested: (a) every distractor shares exactly one feature with the target object, or, (b) every distractor shares exactly two features with the target. Both cases resulted in sequential search, but the slope in (b) was always steeper than the slope in case (a). These results are consistent with SWIFT. In case (a), on average the minimal feature will eliminate *2/3* of the distractors. In (b), only *1/3* would be eliminated on average. Thus SWIFT predicts that the slope in (a) should be about half that of (b).

Search asymmetries. There is another search paradigm where constant and linear time searches have been reported. Searching for a line oriented 18^o among vertical lines can be done in constant time, but searching for a vertical line among these oblique lines takes linear time (Treisman 1988). This asymmetry is explained by assuming that the early representation includes a finite number of orientations that are coarse coded, including vertical and an orientation greater than 18^o. Each oblique line is represented as a combination of activity in the vertical map and the map coding a successive orientation. If this is true, then a pattern containing a single oblique line among a field of vertical lines will cause several regions of activity in the vertical map but only a single region of activity in the other map. The presence of the oblique line can therefore be detected in constant time by computing a global OR. However, the image of a vertical line among several oblique lines will generate several active regions in both maps except at one location, where only the vertical map is activated. In this case, the network must bind the presence of activity in one map with the absence of activity at the same location in another map. This requires serial search.

Similar asymmetries are present when detecting curvature, circles vs ellipses, single vs paired lines, etc. In all of these cases, a central question is: how does the brain know what to do? The subject has no knowledge about his/her internal representations. Just knowledge about the target object is insufficient - the map that is searched depends on the particular image. The answer is simple if SWIFT is used: searching the map with the least total activity will always produce the correct results.

Parallel processing of conjunctions. Some authors have reported conjunctive feature searches which always result in flat slopes. (McLeod, Driver, & Crisp 1988) report that the detection of a moving X among static X's and moving O's can be done in parallel. (Nakayama & Silverman 1986) tested conjunction searches using the features color, motion, and depth. They found that motion-color conjunctions required serial processing, whereas depth-color and depth-motion conjunctions could be processed in parallel.

Recently (Treisman & Sato 1990) and (Wolfe, Cave & Franzel 1989) have suggested models where conjunctions can be detected in constant time with top-down information. It is possible to implement Treisman and Sato's Feature Inhibition model in our architecture. They suggest that if the features that are *not present* in the target inhibit the priority map (i.e. P_f is negative) then a location containing the conjunction of two features would retain the highest priority. This can be easily modeled in our network, however there is one problem: it cannot explain sequential search! If people can use such a general strategy, why do we get linear search times at all? A related problem is that both models cannot explain why only specific feature combinations give rise to parallel search.

SWIFT can explain these results if one assumes that certain feature combinations are represented in parallel. For example, (McLeod, Driver, & Crisp 1988) mentions that area MT contains cells which are tuned to both direction of motion and orientation. Since a primary feature that distinguishes X's from O's is an oriented line, a moving X should produce a unique pattern of activity in this feature map. If such combinations are present, then SWIFT would select the appropriate feature map and detect the target in constant time.

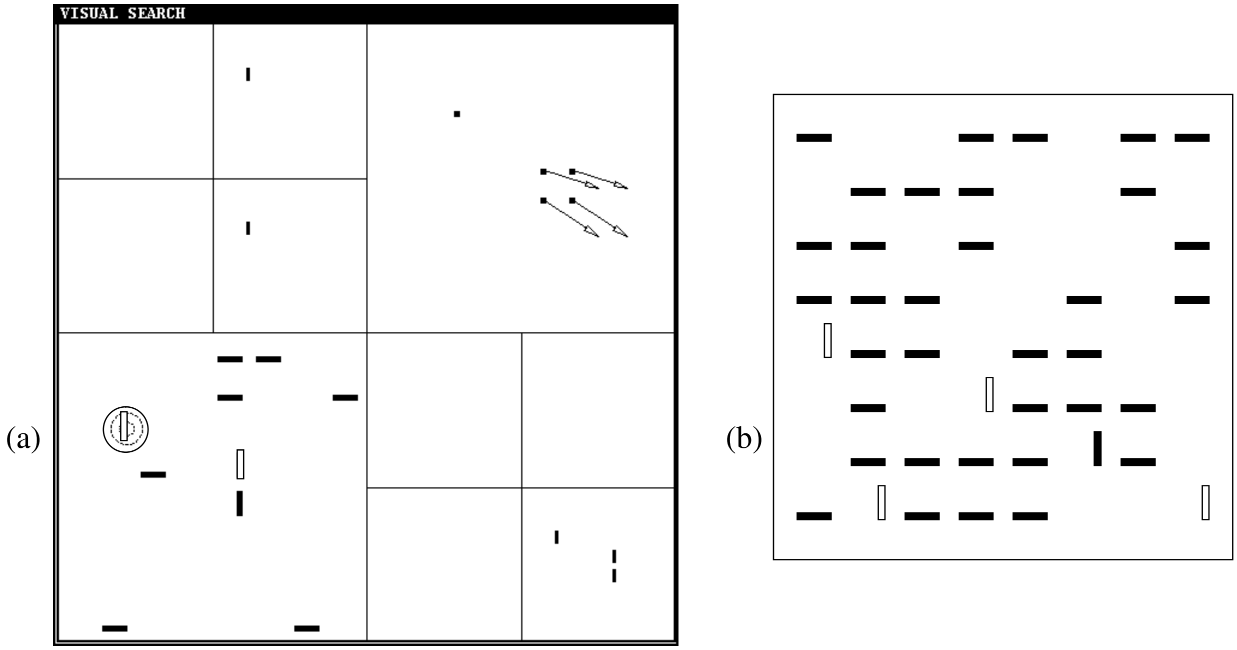

Figure 3. (a) Sample output from our simulator. (b) An image with M=5. (No shading => red, filled black => blue.)

Concluding Remarks

Optimal features for visual search. In light of the above results it is natural to ask what the best set of features should be. If SWIFT is used as a constraint, then we want the set of features that minimize *M* over all possible images and target objects, i.e. that best discriminate objects. It is easy to see that the optimal set of features should be maximally uncorrelated and that the distribution of feature values should be uniform over the space of possible objects. In other words, the optimal features should be the principal components of the distribution of images. It is interesting to note that a single Hebb neuron extracts the largest principal component of the input distribution and with inhibition, sets of Hebbian neurons can extract successively smaller components. Moreover, as some researchers have demonstrated (e.g. Linsker 1989), simple Hebbian learning can lead to features that look very similar to the features in the visual cortex. If the early features in visual cortex are in fact the principal components, then SWIFT is a simple strategy that takes advantage of it.

Other computational models. (Chapman 1990) has implemented a pyramid model of attention. It has only been used to replicate the original single feature vs conjunctive feature searches although, in principle, a control strategy like SWIFT could be employed. (Wolfe, Cave, & Franzel 1989) have simulated a model of visual search which they call the Guided Search model. Their model accounts for a wider range of results than Chapman's. Our model is consistent with their philosophy in that a smart parallel strategy is used to rank possible candidates. However their model cannot account for the search asymmetry results or the *2:1* ratios in the slopes. In addition, the model requires complete connectivity of the units in the feature maps. This would require $O(N^2)$ weights (resulting in approximately 10^{12} connections for a 1000x1000 image) and is therefore not implementable for high-resolution images. Neither of these models implement a continuous focus of attention: each object is assumed to occupy exactly one pre-determined location.

To our knowledge, the only realistic implementation of visual search that works with pixel-based images is in (Mozer, 1991). The model also implements a continuous focus. There are some differences in our models. In our model every aspect of the control process is made explicit, including the use of top-down information. In their model the differing search slopes are explained by assuming a specific amount of noise in the activations of the feature maps. In our model search time depends on the feature representations and the minimal feature map.

Conclusions. We have presented efficient psychologically plausible connectionist mechanisms for visual attention. These mechanisms have been integrated into a complete system for visual search. The resulting network scales well both in terms of the number of connections (linear in the number of pixels) and in the focusing time (constant). The implementation of a single plausible search strategy, SWIFT, was shown to be consistent with the single/conjunctive search, the 2:1 ratio in the target absent/present slopes, and dependence on localization. The strategy extends other sequential integration models in that it is also consistent with search for triple conjunctions, search asymmetries, search within a feature, and possibly the constant time detection of certain feature combinations.

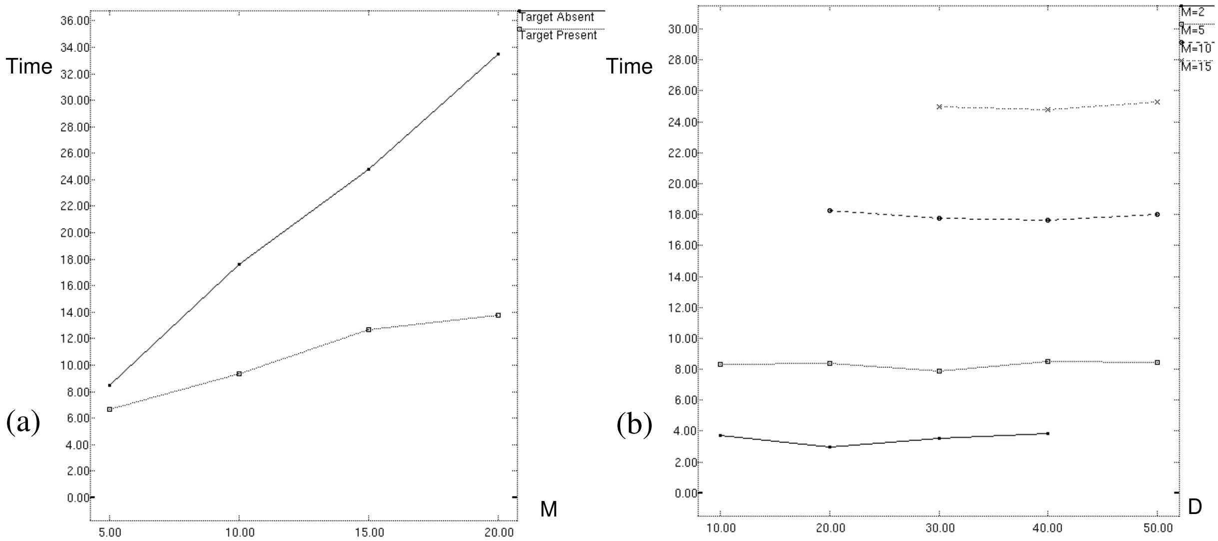

Figure 4. In (a), D is constant at 40 while M is varied. In (b) M is kept fixed and D is varied. (Target absent only. Target present results are similar.)

Acknowledgments

We thank Peter Blicher, Jerome Feldman, Jitendra Malik, and especially Anne Treisman for their helpful comments and stimulating discussions.

References

Ahmad, S., and Omohundro, S. 1990a. Equilateral Triangles: A Challenge for Connectionist Vision. In: Proceedings of the 12th Annual Conference of the Cognitive Science Society, MIT, July, 1990.

Ahmad, S., and Omohundro, S. 1990b. A Connectionist System for Extracting the Locations of Point Clusters, Technical Report, TR-90-011, International Computer Science Institute, Berkeley, CA.

Chapman, D. 1990. *Vision, Instruction, and Action*. Ph.D. diss., Dept. of Computer Science, MIT.

Egeth, H.E., Virzi, R.A., and Garbart, H. 1984. Searching for Conjunctively Defined Targets. *Journal of Experimental Psychology: Human Perception and Performance*, **10**(1):32-39.

Julesz, B., and Bergen, J.R. 1987. In: Fischler, M.A. and Firschein, O. (Eds.) *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Morgan Kaufmann.

Linsker, R. 1989. How to Generate Ordered Maps by Maximizing the Mutual Information Between Input and Output Signals. *Neural Computation*, **1**:402-411.

McLeod, P., Driver, J., and Crisp, J. 1988. Visual search for a conjunction of movement and form is parallel. *Nature*, **332**:154-155.

Mozer, M. 1991. *The Perception of Multiple Objects: A Connectionist Approach*. MIT Press, Cambridge, MA.

Nakayama, K., and Silverman, G. 1986. Serial and parallel processing of visual feature conjunctions. *Nature*, **320**:264-265.

Quinlan, P.T. and Humphreys, G.W. 1987. Visual search for targets defined by combinations of color, shape, and size: An examination of the task constraints of feature and conjunction searches. *Perception & Psychophysics*, **41**:455-472.

Rock, I., Linnett, C.M., Grant, P., and Mack, A. 1990. Results of a New Method for Investigating Inattention in Visual Perception. Paper presented at 31'st annual meeting of the Psychonomic Society, New Orleans, LA, November, 1990.

Sejnowski, T.J. 1986. Open Questions About Computation in Cerebral Cortex. In McClelland, J.L., and Rumelhart, D.E. (Eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA, MIT Press.

Sparks, D. L. 1986. Translation of Sensory Signals into Commands for Control of Saccadic Eye Movements: Role of Primate Superior Colliculus, *Physiological Reviews,* **66**(1).

Treisman, A. 1988. Features and Objects: The Fourteenth Bartlett Memorial Lecture. The Quarterly Journal of Experimental Psychology, **40A** (2).

Treisman, A., and Sato, S. 1990. Conjunction Search Revisited. Journal of Experimental Psychology: Human Perception and Performance, **16**(3):459-478.

Wolfe, J.M., Cave, K.R., and Franzel, S.L. 1989. Guided Search: An alternative to the modified feature integration model for visual search. *Journal of Experimental Psychology: Human Perception and Performance*, **15**:419-433.

Yantis, S., and Jonides, J. 1990. Abrupt Visual Onsets and Selective Attention: Voluntary Versus Automatic Allocation. *Journal of Experimental Psychology: Human Perception and Performance*, **16**(1):121-134.