A Maximum-Likelihood Strategy for Directing Attention during Visual Search

Hemant D. Tagare, *Member*, *IEEE*, Kentaro Toyama, and Jonathan G. Wang

Abstract—A precise analysis of an entire image is computationally wasteful if one is interested in finding a target object located in a subregion of the image. A useful "attention strategy" can reduce the overall computation by carrying out fast but approximate image measurements and using their results to suggest a promising subregion. This paper proposes a maximum-likelihood attention mechanism that does this. The attention mechanism recognizes that objects are made of parts and that parts have different features. It works by proposing object part and image feature pairings which have the highest likelihood of coming from the target. The exact calculation of the likelihood as well as approximations are provided. The attention mechanism is adaptive, that is, its behavior adapts to the statistics of the image features. Experimental results suggest that, on average, the attention mechanism evaluates less than 2 percent of all part-feature pairs before selecting the actual object, showing a significant reduction in the complexity of visual search.

Index Terms—Attention, object recognition, visual search.

1 Introduction

Object recognition, or visual search, is the task of finding a known object, called the *target*, in an image. Object recognition algorithms work by selecting regions (or features) of an image and verifying whether the target is present in them. We call the strategy for choosing image regions an "attention strategy."

The attention strategy has significant influence on the speed of the algorithm. A poor attention strategy is likely to investigate many regions of the image that do not contain the object and, thus, waste computation. Conversly, a good attention strategy is likely to choose the right region early on and terminate the search quickly.

A useful attention strategy can be developed as follows: First, perform some computationally inexpensive measurements of the image. Then, use the measurements to reorder the image search so that promising regions are explored first. We explore this idea in this paper. We call the fast measurements *preattentive features.* They are obtained by a *preattentive module* in the recognition system. The subsequent slower and more detailed search is carried out by a *postattentive module*. The attention strategy mediates between the two in order to terminate the search quickly.

Our attention strategy is based on the maximum-likelihood (ML) decision rule. The strategy has two important properties: First, it can be used with different preattentive features (color, edges, corners, etc.) and with different postattentive modules without modification. Second, it adapts to image content. The adaptive nature of the strategy is the key to its success. To understand the adaptive nature, consider the possibility of using color as a preattentive feature. Suppose we are interested in finding an object which has a red part. Then, we might measure pixel color and begin our search on those image regions which have the appropriate hue of red. This will work well unless the image has other red objects besides the target. In that case, it may be better to choose regions with some other color, say blue, provided that target has a blue region but the other objects don't. The point is that, to be effective, the attention strategy should adapt—its behavior should be dependent on the total amount of red or blue in the image. The ML attention strategy demonstrates this behavior, as we show in Section 5.

As we mentioned above, we assume that the recognition system is composed of three subsystems (Fig. 1): 1) a *preattentive system*, 2) an *attention mechanism*, and 3) a *postattentive system*. The preattentive system is a fast feature detector. It operates over the entire image and detects simple image features (color regions, edges, corners, etc.). We call these *preattentive features*. Some of the features come from parts of the target, the rest come from other objects in the image, called *distractors*. We do not assume that the preattentive features are complete or even that they can distinguish the target from the background by themselves.

The role of the attention mechanism is to choose a feature, pair it with a target part, and hypothesize that this pairing is due to the presence of the target in the image. This hypothesis is passed to the postattentive system, which uses full geometric knowledge of the target and explores the image around the feature to find the target. The postattentive system is just a traditional object recognition algorithm. It indicates to the attention mechanism whether the hypothesis is valid or not. If the hypothesis is not valid, then the attention mechanism takes this into account and proposes the next hypothesis. The postattentive system now focuses on a new region of the image. The process terminates either when the target is found or when all features in the image have been exhausted.

There is one subtle aspect of this. It is important that the attention mechanism work with parts of the target rather

- H.D. Tagare is with the Department of Diagnostic Radiology, Department of Electrical Engineering, Yale University, New Haven, CT 06520. E-mail: hemant.tagare@yale.edu.
- K. Toyama is with Microsoft Research, Redmond, WA 98052. E-mail: kentoy@microsoft.com.
- J.G. Wang is with Credit Suisse First Boston, 11 Madison Ave., New York, NY 10010. E-mail: jonathan.wang@csfb.com.

Manuscript received 4 Dec. 1998; revised 11 July 2000; accepted 3 Aug. 2000. Recommended for acceptance by H.I. Christensen. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108397.

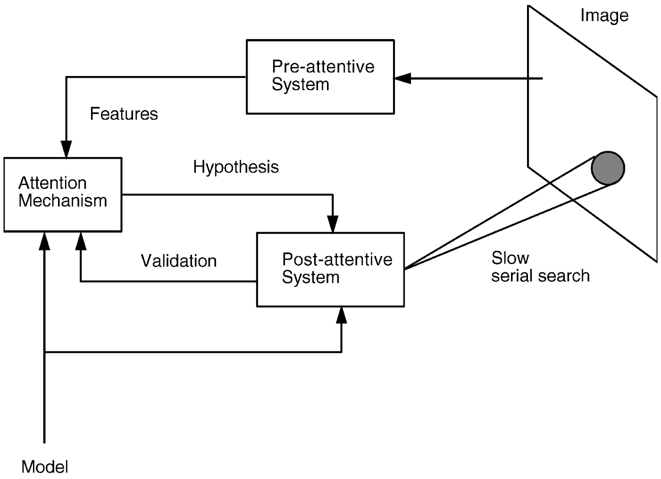

Fig. 1. Attention in visual search.

than the entire target. If the target has many parts, then the attention mechanism has the freedom to choose the part that is used in the pairing. As we shall see in Section 5, this is the reason why the attention mechanism adapts to the image content. If there are many red features in the images and only a few blue features, the attention mechanism chooses to explore the pairing of a blue part with a blue feature rather than a red part with a red feature. Without the capacity to work with parts of the target, the attention mechanism would not have this property.

We also assume that:

1. All features detected by the preattentive system are available to the attention mechanism all the time. This is simply another way of saying that the preattentive system is fast enough to process the entire image before a detailed search with the postattentive system begins.
2. The attention mechanism only uses the values of the features detected by the preattentive system. It does not evaluate geometric constraints between features. Evaluating geometric constraints between pairs, triples, etc., of features is computationally expensive and we wish to avoid this expense.
3. The attention mechanism is greedy. At every stage, it chooses that part-feature pair which has the highest likelihood of coming from the target in the image. The likelihood is evaluated after taking into account all previous pairs rejected by the postattention mechanism.

We make no other assumptions. In particular, we do not assume any specific pre and postattentive systems. The strategy that we derive can be used with a range of user-supplied modules. We demonstrate this in our experiments, where we use corner detectors as well as color detectors as preattentive systems.

In this paper, we only address 2D object recognition and we have kept the formulation as simple as possible. This is by design. We want to emphasize basic ideas and certain approximations. The theory in this paper can be made more sophisticated: by adding more features, by considering multiple spatial resolutions, by clustering features, etc. These alternatives are not pursued here.

1.1 Human Visual Attention

Human vision is known to have an attention strategy that is very effective [14], [22], [28]. Cognitive scientists do not yet have a complete understanding of human visual attention, but some partial understanding has emerged. Many models proposed by cognitive scientists are similar to our model of Fig. 1 [22], [28]. In these models, the preattentive system is fast and capable of extracting "primitive" image features (such as color, edge smoothness, and size). The postattentive system is slower, but can analyze image regions in detail. The human attention mechanism uses primitive features produced by the first system to direct the second.

The behavior of human visual attention in two conditions, called "pop-out" and "camouflage," is particularly interesting. If an image contains a target whose primitive features are sufficiently different from the distractor features, then the time required to find the target is independent of the number of distractors in the image [22]. This is the "pop-out" condition. In this condition, the target just seems to pop out of the image. On the other hand, if all target features are similar to the distractors, then the time required to find the target grows linearly with the number of distractors. We call this the "camouflage" condition.

In Section 5, we show that "pop-out" and "camouflage" are emergent properties of our algorithm. In Section 6, we confirm this experimentally.

1.2 Relation to Previous Work

Other researchers have presented strong cases for using attention in vision algorithms [23], [24]. Some researchers have proposed specific computational mechanisms that model what is known about human visual search [10], [11]. Others have proposed search algorithms for specific cues: parallel-line groups [17], color [5], [18], texture [19], prominent motion [25], "blobs" in scale space [12], or intermediate objects [27]. Some authors have considered attention for scene interpretation [16]. Others have applied it to passive tracking [21] and to active vision systems [2], [15].

Our aim is quite different from these studies. We do not want to implement a specific biology-based attention algorithm or one that is tailored to a particular cue. Instead, we ask whether it is possible to derive an attention strategy from first principles which can be applied to a range of cues.

Finally, we wish to address a source of confusion. Our algorithm is sometimes compared to the *interpretation tree* algorithm for object recognition [1], [6], [7]. The confusion arises from an apparent similarity between the two: Both attempt to match target parts to the image. However, this similarity is only superficial. There are substantial differences between the two algorithms:

1. The two algorithms operate at different levels. We are concerned with interaction between pre- and postattentive modules, rather than organizing the geometric comparison of parts and features. Interpretation trees are concerned with the latter.
2. Interpretation trees attempt to match the entire target to the edges in the image. Our algorithm is concerned with evaluating the likelihood of a single part matching a single feature, given that the target is present in the image.

3. The interpretation tree is an explicitly geometric algorithm—its purpose is to systematically evaluate geometric relations between image edges. On the other hand, our attention mechanism is not concerned with geometric relations. It works for arbitrary feature types, many of which do not enforce any geometric constraints.

These comments are not meant as a criticism of interpretation trees, but are simply meant to show that the two have different goals. In fact, the two can be used, together with the interpretation tree serving as the postattentive system capable of fully recognizing the target.

1.3 Organization of the Paper

This paper is organized as follows: Section 2 contains the definitions and the notation. Section 3 contains the likelihood calculations. The calculation of the exact likelihood is computationally expensive and Section 4 contains approximations to it. Section 5 analyzes the behavior of ML attention strategy and demonstrates its "adaptive" nature. Section 6 contains experimental results and Section 7 concludes the paper.

2 Definitions

2.1 Features and Parts

We begin by defining preattentive features. By a preattentive feature, or simply a *feature*, we mean a primitive element of an image, such as color, corner, etc., that can be found by simple feature detectors. A feature has a value (the RGB triple of color, the angle of a corner, etc.) which belongs to some feature space $\mathcal{V}$. The set of all features in the image is $\mathcal{F}$. We will refer to the value of the $k\mathrm{th}$ feature by f_k. Some of the features in the image come from the instance of the target we want to detect. These are *target features*. Others come from other objects in the image; these are *distractor features*.

The target to be recognized has M *parts*, $S_j, j = 1, \ldots, M$. The set of all parts is $\mathcal{P}$. Parts need not be defined in a geometric way. The only requirement is that the union of all parts is the entire target. A part may be visible or may be completely occluded in an image. The prior probability that part S_j is visible in the image is P_j (the probability of complete occlusion of the part is $1 - P_j$). We assume that each visible part gives rise to a single feature in the image. Thus, multiple parts cannot contribute to a feature and a part cannot give rise to multiple features. If a part is visible, it may still be partially occluded and its feature value may change due to partial occlusion. We model this by saying that, if the $j\mathrm{th}$ part is visible, then its feature value is a random variable with the probability density function $p_j(f)$.

We assume that distractor feature values are realizations of a uniform Poisson process in the feature space $\mathcal{V}$. We make this assumption because we do not have any knowledge of distractors and would like to treat their values as being "uniformly" distributed in $\mathcal{V}$. The probability density of obtaining n distractor feature values $f_1, \cdots, f_n$ is given by

$$p_d(f_1, \cdots, f_n) = \frac{(\lambda V)^n e^{-\lambda V}}{n!}, \qquad (1)$$

where λ is the process intensity and V is the feature space volume.

2.2 The Attention Mechanism

The attention mechanism is iterative and it works as follows: During each iteration, the mechanism chooses that part-feature pair which is most likely due to the target. This choice is passed on to the postattentive system, which evaluates whether the pairing is really due to the occurrence of the target. If it is due to the target, then the target has been found and the search terminates. If it is not, then the attention mechanism takes this information into account and suggests the next most likely pair.

We will denote the pairing of part S_m with feature f_n as (S_m, f_n). Since the set of all parts of the target is $\mathcal{P}$ and the set of all image features is $\mathcal{F}$, the set of all possible part-feature pairings is $\mathcal{P} \times \mathcal{F}$. We will refer to any pair (S_i, f_k) which has been declared incorrect by the postattention mechanism as a *rejected pair*. The set of all rejected pairs until the $j\mathrm{th}$ iteration of the algorithm is denoted by R^j. Thus, in the $j\mathrm{th}$ iteration, the set of all part feature pairs that have not been rejected is $\mathcal{P} \times \mathcal{F} - R^j$. With this notation, the pseudocode for the attention algorithm is:

```
1. Preprocess: Extract $\mathcal{F}$, the set of image features.
2. Initialize: Set $j = 1$, $R^1 = \emptyset$, the empty set.
3. Loop condition: The set of pairs that remain to be tested at the $j\mathrm{th}$ iteration is
   $\mathcal{P} \times \mathcal{F} - R^j$. If this set is empty, terminate the iteration and declare that the target
   is not present in the image.
4. Candidate selection: From the set $\mathcal{P} \times \mathcal{F} - R^j$ choose the pair $(S_{m(j)}, f_{n(j)})$ which has the
   greatest likelihood of coming from the target in the image:

   $(S_{m(j)}, f_{n(j)}) = \arg \max_{(S_m, f_n) \in \mathcal{P} \times \mathcal{F} - R^j} p((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j),$   (2)

   where $p((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j)$ is the likelihood that the pair $(S_m, f_n)$ comes from the target
   in the image given the set of parts $\mathcal{P}$, the set of image features $\mathcal{F}$, and the set of
   rejected pairs $R^j$.
   This is the ML decision.
5. Object verification: Pass the selected pair to the postattentive system for verification.
   If the hypothesis is correct, the target has been found. Terminate the search.
   Else ...
6. Bookkeeping: Set $R^{j+1} = R^j \cup \{(S_{m(j)}, f_{n(j)})\}$. Set $j = j + 1$. Go to Step 3.
```

3 The Likelihood

We need a formula for $p((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j)$ to execute the above algorithm. We begin with a simple calculation.

3.1 All Parts Visible

Assume for the moment that there are no rejected pairs and that all parts of the target are visible, i.e., the prior probability $P_j = 1$ for all j. Suppose that there are N features in the feature set $\mathcal{F}$. We first evaluate the likelihood that a specific set of M features came from the M target parts, with the rest of the features being distractors. To describe the pairing of parts with features, we introduce a *part mapping function* $\pi : \{1, \cdots, M\} \rightarrow \{1, \cdots, N\}$ ($\pi(i) \neq \pi(j)$, if $i \neq j$) from the indices of the parts to the indices of features. The function says that the parts $S_1, \cdots, S_M$ are mapped to features $f_{\pi(1)}, \cdots, f_{\pi(M)}$. The likelihood of this with the rest of the feature set accounted for as distractors is:

$$\left(\prod_{i=1,\cdots,M} p_i(f_{\pi(i)}) \right) \times p_d\big(\mathcal{F} - \{f_{\pi(1)}, \cdots, f_{\pi(M)}\}\big).$$

Expressions such as the above occur frequently in our analysis and we use a special notation for them. We denote the expression by $\phi(g_1, \cdots, g_M, H)$,

$$\phi(g_1, \cdots, g_M, H) = \left(\prod_{i=1,\cdots,M} p_i(g_i) \right) \times p_d(H),$$

where $g_1, \cdots, g_M$ are the features to be matched with $S_1, \cdots, S_M$, respectively, and H is the set of distractor features. In this notation, the above likelihood is $\phi(f_{\pi(1)}, \cdots, f_{\pi(M)}, \mathcal{F} - \{f_{\pi(1)}, \cdots, f_{\pi(M)}\})$.

Now, the likelihood that the single pair (S_m, f_n) comes from the target is the sum of all likelihoods in which the parts are paired with features with the restriction that part S_m is always paired with part f_n. This is

$$\sum_{\pi, \pi(m)=n} \phi\big(f_{\pi(1)}, \cdots, f_{\pi(M)}, \mathcal{F} - \{f_{\pi(1)}, \cdots, f_{\pi(M)}\}\big),$$

where the over all sum is π functions which satisfy $\pi(m) = n$.

Next, suppose that the set of rejected pairs R^j is not empty. To calculate the likelihood that (S_m, f_n) is due to the target, we must avoid summing over those part mappings which give rise to a rejected part-feature pair. We will say that a part mapping $\pi : \{1, \cdots, M\} \rightarrow \{1, \cdots, N\}$ is *compatible* with the set R^j if $(S_i, f_{\pi(i)}) \notin R^j$ for $i = 1, \cdots, M$. Using this notion, we can write the likelihood $p((S_m, f_n), \mathcal{P}, \mathcal{F}, R^j)$ as

$$p\big((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j\big) = \sum_{\pi, \pi(m)=n} \phi\big(f_{\pi(1)}, \cdots, f_{\pi(M)}, \mathcal{F} - \{f_{\pi(1)}, \cdots, f_{\pi(M)}\}\big), \tag{3}$$

where the sum is over all part of mapping functions that are compatible with R^j and which satisfy, $\pi(m) = n$.

3.2 Occluded Parts

Next, consider the possibility that some parts may be completely occluded (the prior probabilities P_j are not necessarily equal to 1). To take this into account, we introduce additional features called *null features*. When a part is mapped to a null feature, we say that it is completely occluded.

We augment the feature set $\mathcal{F}$ by adding M null features to it. The feature set now has $N + M$ elements. The likelihood $p((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j)$ can be expressed as before:

$$p\big((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j\big) = \sum_{\pi, \pi(m)=n} \phi\big(f_{\pi(1)}, \cdots, f_{\pi(M)}, \mathcal{F} - \{f_{\pi(1)}, \cdots, f_{\pi(M)}\}\big), \tag{4}$$

where, as before, the sum is over all compatible part mapping functions, but the function ϕ is now given by

$$\phi(\{g_1, \cdots, g_M, H\}) = \left\{ \prod_{j=1,\cdots,M} q_j(g_j) \right\} \times h(H), \tag{5}$$

in which q_j and h evaluate the likelihood that feature values come from parts and from distractors taking into account the prior probability of occlusion and null features:

$$q_j(g_j) = \begin{cases} P_j p_j(g_j) & \text{if } g_j \text{ is not a null feature} \\ (1 - P_j) & \text{if } P_j \text{ is a null feature} \end{cases} \tag{6}$$

and

$$h(H) = \frac{(\lambda V)^{N_1} e^{-\lambda V}}{N_1!}, \tag{7}$$

where

$$N_1 = \text{number of nonnull features in } H.$$

So far, we have ignored the fact that we do not know the intensity λ of the distractor process (λ is required in (7)). However, we can estimate l from the data as follows: Since $P_i, i = 1, \ldots, m$ is the probability that part S_i is visible, the average number of visible parts is $\sum_i P_i$. Thus, the average number of distractors is $N - \sum_i P_i$ and this number must be equal to λV, which is the average number of distractors derived from the Poisson distribution. That is,

$$\lambda V = N - \sum_i P_i$$

or

$$\lambda = \frac{\left(N - \sum_i P_i\right)}{V}. \tag{8}$$

Equations (4), (5), (6), (7), and (8) completely define the likelihood.

4 Approximations to the Likelihood

Equations (4), (5), (6), (7), and (8) are computationally expensive to evaluate because they contain the combinatorics of mapping $M - 1$ parts to $N - 1$ features. In this section, we propose two approximations. The first involves using the normal approximation to the Poisson distribution. This allows us to simplify the expression for the likelihood by eliminating some terms. The second, and the more radical approximation, involves using a reduced number of parts. That is, we consider smaller targets formed by taking r-tuples of parts from the original target and we calculate the likelihood that a part-feature pair comes from at least one of the simpler targets.

We find that the simple case of $r = 2$ gives very satisfactory results in practice and we use this approximation in all our experiments.

4.1 The Normal Approximation

The Poisson distribution can be approximated by a normal distribution when the number of distractors is large [8]. Recall that each term being summed in (4) has a factor $h(H)$ arising from (5). In the Appendix, we show that if $\mathcal{N}$ is the number of null features mapped onto the parts, then $h(H)$ can be approximated as:

$$\begin{aligned} h(H) &\simeq \frac{1}{2\pi\sqrt{N-\sum_i P_i}} \exp\left\{-\frac{M^2}{2N} - \frac{M\sum_i P_i}{N}\right\} \prod_{j=1}^{\mathcal{N}} \exp\left\{\frac{M}{N}\right\}, \\ &= C \prod_{j=1}^{\mathcal{N}} \exp\left\{\frac{M}{N}\right\}, \end{aligned} \tag{9}$$

where, C is the part of the expression that is independent of $\mathcal{N}$.

Referring back to (5), we can see that the term C is a common factor in all ϕ terms and need not be evaluated if we are only interested in finding the part-feature pair that maximizes the likelihood of (4). With this, the likelihood becomes:

$$\begin{aligned} &p\big((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j\big) = \\ &\sum_{\pi,\pi(m)=n} \phi\big(f_{\pi(1)}, \cdots, f_{\pi(M)}, \mathcal{F} - \{f_{\pi(1)}, \cdots, f_{\pi(M)}\}\big), \end{aligned} \tag{10}$$

where the function ϕ is now given by

$$\begin{aligned} &\phi(\{g_1, \cdots, g_M\}, H) = \\ &\left\{\prod_{j=1,\cdots,M} q_j(g_j)\right\} \times \prod_{j=1}^{\mathcal{N}(g_1,\cdots,g_M)} \exp\left\{\frac{M}{N}\right\} \times C = \prod_{j=1,\cdots,M} r_j(g_j), \end{aligned} \tag{11}$$

where $\mathcal{N}(g_1, \cdots, g_M)$ is the number of null features in $g_1, \cdots, g_M$ and

$$r_j(g_j) = \begin{cases} P_j p_j(g_j) & \text{if } g_j \text{ is not a null feature} \\ (1-P_j)\exp\left(\frac{M}{N}\right) & \text{if } g_j \text{ is a null feature.} \end{cases} \tag{12}$$

In the last step of (11), we dropped the C term and included the $\exp\left(\frac{M}{N}\right)$ term in the definition of r_j.

4.2 Matching Simpler Models

We now proceed to the second approximation. In this approximation, we consider all simpler targets formed by the part S_m and $r-1$ tuples of other parts and evaluate the likelihood that (S_m, f_n) comes from at least one the simpler targets.

4.3 Approximate Likelihood for $r = 1$

For $r = 1$, we have M simplified targets, each being exactly one of the parts. The likelihood that the pair (S_m, f_n) comes from at least one of these simpler targets is equal to the likelihood that it comes from the target having S_m as its single part. The likelihood is just

$$r_m(f_n). \tag{13}$$

4.4 Approximate Likelihood for $r = 2$

In this case, we have M simplified targets, each target having two parts—the part S_m and one other part and we calculate the likelihood that the pair (S_m, f_n) comes from at least one of these simpler parts.

Let S_i $(i \neq m)$ be another part. Then, using (10), (11), and (12), the joint likelihood that the pair $(S_m, f_n) \notin R^j$ comes from this simplified target is

$$\sum_{\pi,\pi(m)=n} r_m(f_{\pi(m)}) r_i(f_{\pi(i)}),$$

where, the sum is over all part mapping functions π which 1) map the part index set $\{m, i\}$ into the feature index set, 2) are compatible with R^j, and 3) satisfy $\pi(m) = n$. The functions r_m, r_i are defined by (12). This expression can be easily rewritten as

$$\sum_{\pi,\pi(m)=n} r_m(f_{\pi(m)}) r_i(f_{\pi(i)}) = r_m(f_n) \left(\sum_{k,k\neq n,(S_i,f_k) \notin R^j} r_i(f_k) \right),$$

where, on the righthand side, the sum is over all feature indices k for which $k \neq n$ and the pair (S_i, f_k) is not in R^j.

Therefore, the joint likelihood that the pair (S_m, f_n) comes from one or more of the 2-part targets is the sum of the above likelihood over all $i, i \neq m$:

$$p\big((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j\big) = r_m(f_n) \sum_{i\neq m} \left(\sum_{k,k\neq n,(S_i,f_k) \notin R^j} r_i(f_k) \right). \tag{14}$$

The computational complexity of this expression is $O(NM)$.

4.5 Higher Order Approximations

Now, consider the likelihood that the pairing $(S_m, f_n) \in \mathcal{P} \times \mathcal{F} - R^j$ is due to at least one simplified target formed by the part S_m and $r-1$ other parts of the original target.

The expression for this likelihood is messy. To simplify its presentation, we adopt the following convention: We let $I = \{i_1, \cdots, i_r\}$ represent an ordered set of r indices such that $\{S_{i_1}, \cdots, S_{i_r}\}$ is one simplified target. Two different sets of the type I represent two different combinations of r parts from the target.

Repeating the calculation for (14), we get

$$p\big((S_m, f_n) \mid \mathcal{P}, \mathcal{F}, R^j\big) = \sum_{I, i_m = m} \sum_{\pi,\pi(i_m)=n} \left(\prod_{i \in I} r_i(f_{\pi(i)}) \right), \tag{15}$$

where the first sum is over all I which have $i_m = m$, the second sum is over all part mapping functions π (which map I to the feature index set) which are compatible with R^j and for which $\pi(i_m) = n$. The complexity of evaluating the likelihood of (15) is $O(C_{r-1}^{N-1})$.

The calculations in (13), (14), and (15) give us likelihoods which can, in practice, be used with the attention algorithm. This completes the description of the algorithm. Next, we turn to investigate the adaptability of the attention algorithm.

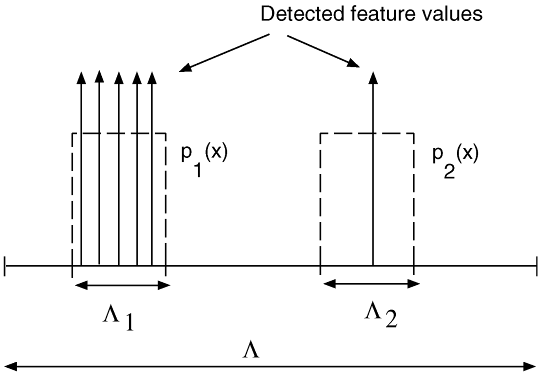

Fig. 2. Maximum-likelihood explanation of pop-out.

5 Adaptation, Pop-Out, and Camouflage

5.1 Adaptation

To understand the adaptive behavior of the algorithm, consider the following simple case:

1. The model has only two parts, S_1 and S_2. Neither part is ever completely occluded (i.e., $P_1 = P_2 = 1$).
2. The feature space is one-dimensional and the parts have uniform feature distributions over disjoint intervals, Λ_1 and Λ_2, respectively, of equal lengths (length = L). That is, the probability density that S_1 occurs with a feature value f is

$$p_1(f) = \begin{cases} \frac{1}{L} & \text{if } f \in \Lambda_1 \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, the probability density that S_2 occurs with a feature value f is

$$p_2(f) = \begin{cases} \frac{1}{L} & \text{if } f \in \Lambda_2 \\ 0 & \text{otherwise.} \end{cases}$$

Consider two situations in which there are six features in the image. In the first case, five of the six features, $f_1, \cdots, f_5$, occur in Λ_1 and the sixth feature, f_6, occurs in Λ_2 (illustrated in Fig. 2). In the second case, the situation is reversed. The five features, $f_1, \cdots, f_5$, occur in Λ_2, and the sixth feature, f_6, occurs in Λ_1.

In the first case, the likelihood that S_2 matches f_6 is a sum of five terms. Each term corresponds to S_2 matching f_6, with S_1 matching one feature in $\{f_1, \cdots, f_5\}$, and the rest of the features being distractors. Each term is

$$\frac{1}{L^2}\, p_d(4)$$

and, hence, the likelihood of (S_2, f_6) is

$$\frac{5}{L^2}\, p_d(4).$$

Now, consider the likelihood of (S_1, f_1). It has a single term corresponding to S_1 matching f_1, S_2 matching f_6, and $f_2, \cdots, f_5$ being distractor features. Its likelihood is

$$\frac{1}{L^2}\, p_d(4).$$

The likelihood that S_1 matches any other feature occurring in Λ_1 is the same as this.

Clearly, the likelihood that S_2 matches f_6 is greater than the likelihood that S_1 matches f_1 or any other feature in Λ_1 and the attention mechanism chooses the former.

We can simply repeat the above calculation when the five features $f_1, \cdots, f_5$ are in Λ_2 and feature f_6 is in Λ_1. Again, we get the likelihood of part S_1 matching the feature f_6 as

$$\frac{5}{L^2}\, p_d(4),$$

while the likelihood of S_2 matching any of the features in Λ_2 is

$$\frac{1}{L^2}\, p_d(4).$$

The former is clearly greater than the latter.

If Λ_1 and Λ_2 were ranges of red and blue colors, then the two cases can be interpreted as follows: The target has two parts, one colored red and the other blue. In the first case, we have an image with one blue feature and five red features and, in the second case, we have an image with one red feature and five blue features. The attention algorithm chooses to investigate the blue feature in the first case and the red feature in the second case. Thus, the algorithm "adapts" to the distribution of features in the image and chooses to investigate that feature which is least like the distractor features. This is precisely the behavior we want for the attention algorithm and the above calculation shows that the ML decision imparts it to our algorithm.

It is easy to check that, if we use the $r = 1$ approximation in the above calculation, the algorithm *will not* adapt. In fact, it is easy to check, in general, that adaptation is possible if $r \geq 2$.

Finally, recall that pop-out and camouflage are conditions under which the human visual system finds the target in constant time and in time that grows linearly with the number of distractors. Pop-out is achieved if the target has some feature that is sufficiently different from the distractors. Camouflage occurs when all target features are similar to distractor features. The adaptive behavior discussed above demonstrates pop-out since the ML attention mechanism always chooses that feature which is least like the distractors. In contrast, if the target and distractor had similar features—say that there were three features in Λ_1 and Λ_2—the likelihoods of all part-feature pairs would be identical and there would be no reason to prefer one over the other. In this case, the search would proceed without any strong bias toward choosing a particular pair and the time to find the target will be similar to a blind serial search. It will grow linearly with the number of distractors. This is camouflage.

Thus, it appears that the ML attention mechanism can emulate pop-out and camouflage.

6 Experimental Results

Having investigated approximations and properties of the ML attention mechanism we now report its performance in experiments with real images. We have three aims for the experiments: First, we want to demonstrate the performance of the ML attention strategy in real world experiments. Second, we want to demonstrate that pop-out and

TABLE 1
Summary of Experimental Conditions

Exp.	Aim	Target	Distractors	Pre-attentive Feature	Attention Rule
1	Performance Evaluation	Fig. 3	Random Objects from Lab.	Corners + Arms	ML $r = 1, 2, 3$
2	Pop-out & Camouflage Demo	Fig. 3	Similar & Dissimilar Objects	Corners + Arms	ML $r = 2$
3	Alternate Pre-attentive Features	Ronaldo	From existing images	Color	ML $r = 2$
4	Alternate Pre-attentive Features	Waldo	From existing images	Color	ML $r = 2$

camouflage occur with the ML strategy. Finally, we want to show that the attention strategy can be used with different preattentive features.

We conducted four sets of experiments. Table 1 summarizes the salient features of all four. Real images were used in all experiments. In the first two experiments, the preattentive features were corners + their arms (described in detail below). In the last two experiments, the preattentive features were RGB values of colors. The ML attention mechanism was used with all prior probabilities P_j set to 1 and with the r-tuple approximation for $r = 1, 2, 3$. It was easily determined in the first experiment that the approximation with $r = 2$ was the best compromise between speed, efficiency, and adaptability and $r = 2$ was adopted for all subsequent experiments. In all experiments, the postattentive system was a manual check—that is, each part-feature hypothesis suggested by the attention mechanism was manually evaluated for correctness and the decision was fed back to the attention mechanism.

For each image in the experiment, we calculated the total number of part-feature pairings (the size of the set $\mathcal{P} \times \mathcal{F}$). The ratio of the number of part-feature pairings examined until the target was found to the total number of part-feature pairings was taken to be the performance measure for the attention mechanism. Finally, for visual display, the consecutive features in the part-feature pairs suggested by the attention mechanism were plotted on the image and joined by arrows (for example, see Fig. 6). This gives a vivid visualization of the decisions made by the attention mechanism.

The details for each experient are as follows.

6.1 Experiment 1: Performance Evaluation

The aim of this experiment is to evaluate the performance of the attention mechanism. The target used in the experiment is shown in Fig. 3—a cardboard cut-out of a fish. Fifty images containing the target were produced by placing the model in a 30 cm x 30 cm area. Commonly occurring laboratory tools were tossed on the model. Images were taken in such a way that the model had a scale range between 0.5 and 2.0. In all cases, the model was partially occluded in the image. The model was so heavily occluded in two of the 50 images that none of its corners were visible. These images were discarded and the algorithm was tested on the remaining 48 images. Fig. 4 shows two of the 48 images.

The preattentive features were taken to be corners + arms of the edge contours in the image. Corners were defined as points of local maxima of curvature of the edge contour. The two arms of the corner are the edge paths from the corner to the previous and next corner of the contour (Fig. 5). Corners and arms were extracted from the images automatically by edge detection followed by edge linking and curvature calculation. Each corner + arm feature was parameterized by a vector of two parameters: the length of the shorter arm, L, and the average angle of deviation between the two arms, θ (see Fig. 5).

The target was easily seen to have six corners. Each corner of the target and its arms constitutes a part and gives rise to one feature in the image. The distribution of feature values for each part was calculated as follows: Each target corner was occluded (in software) such that the smaller arm length after occlusion was 100 percent, 80 percent, and 60 percent of the length before occlusion. For each partial occlusion, the feature value (θ, L) was calculated. These values represent samples of the distribution of (θ, L) under partial occlusion at unit magnification. The process was duplicated by changing the magnification (in software) of the model to $2.0,\ 1.707,\ 1,\ 0.8,\ 0.5$. The set of (θ, L) obtained in this way was fed to a standard nonparametric density estimator to obtain the probability distribution of corner parameters for each part.

Results. That approximate ML decision rule was used with r set to 1, 2, and 3, respectively. Fig. 6 shows a typical result of ML visual search with $r = 2$. The figure shows the sequence of corners that the algorithm analyzed in turn until it suggested the target. The corner at the successful match is also shown in the figure. Similar behavior was obtained for $r = 1$ and $r = 3$.

As mentioned above, to evaluate the effectiveness of the ML attention mechanism, we measured the average

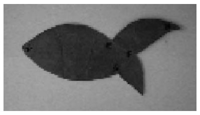

Fig. 3. The model.

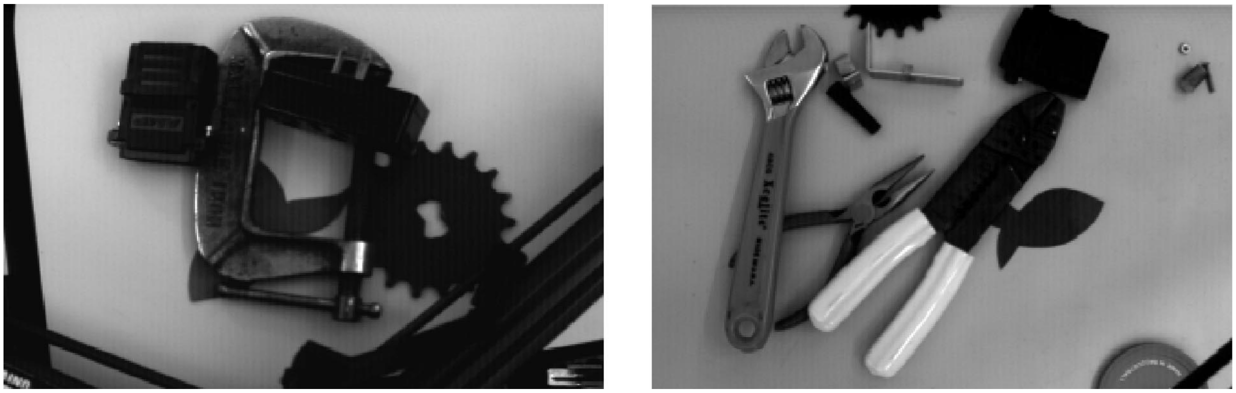

Fig. 4. Example images.

percentage of hypotheses that were processed until the correct hypothesis was suggested. On average, the $r = 1$ approximation processed 4.67 percent of the possible hypotheses, the $r = 2$ approximation processed 1.97 percent of the possible hypotheses, and the $r = 3$ approximation processed 2.73 percent of the possible hypotheses. Without an attention mechanism, on average 50 percent of the possible hypotheses would have to be processed to find the target. Clearly, all three approximations work well, but the latter two outperform the first. Since the $r = 1$ approximation does not exhibit adaptation (as discussed in Section 5), it was dropped from further consideration. Further, since $r = 2$ and $r = 3$ performed similarly, but the $r = 3$ approximation was slower in execution, the $r = 2$ approximation appears to be a good compromise between effectiveness, ability to adapt, and computational complexity. Fig. 7 shows a histogram of the percentage of hypotheses processed by this approximation to find the correct match. The $r = 2$ approximation was adopted for all subsequent experiments.

6.2 Experiment 2: Pop-Out and Camouflage

In the second experiment, we examined the performance of the ML attention mechanism under pop-out and camouflage conditions. As mentioned above, we used the approximate likelihood with $r = 2$.

To simulate pop-out and camouflage conditions, we created similar and dissimilar distractors. Dissimilar distractors were triangular pieces of cardboard. Similar distractors were created by duplicating the target model and cutting the duplicates in half along random lines. Fig. 8 shows an image where the model is present along with the triangular dissimilar distractors. This is the pop-out condition. Fig. 9 shows the camouflage condition. Multiple images were obtained in the pop-out and camouflage conditions by increasing the number of distractors.

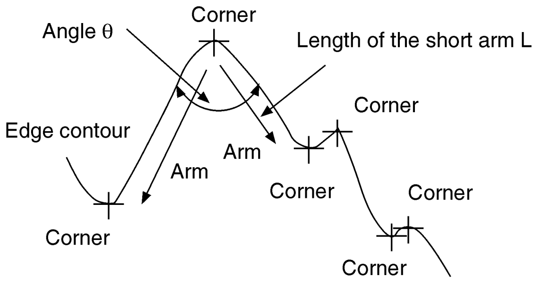

Fig. 5. Features used in experiments.

Figs. 8 and 9 also show typical sequences of image features that were searched until the target was suggested in pop-out and camouflage conditions. Fig. 10 shows the number of hypotheses processed until the target was found, as a function of the total number of features in the image. In the pop-out case, the first hypothesis was always the correct one, while, in the camouflage case, the average number of hypotheses increased monotonically with the number of features in the image.

6.3 Experiment 3: Preattentive Color Features

The aim of the third experiment was to evaluate the performance of the attention mechanism with an alternate preattentive feature. In this experiment, the target was Luiz Ronaldo, a member of the Brazilian soccer team. The preattentive feature was the RGB values of color of pixels within a region. The target was assumed to have two parts corresponding to the yellow and blue of the team's uniform. Color distributions were estimated by taking a sample of each color from several images of the subject,

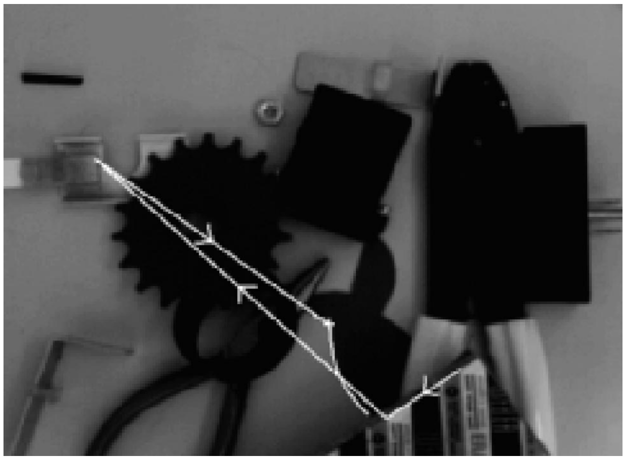

Fig. 6. Attentive search for the object.

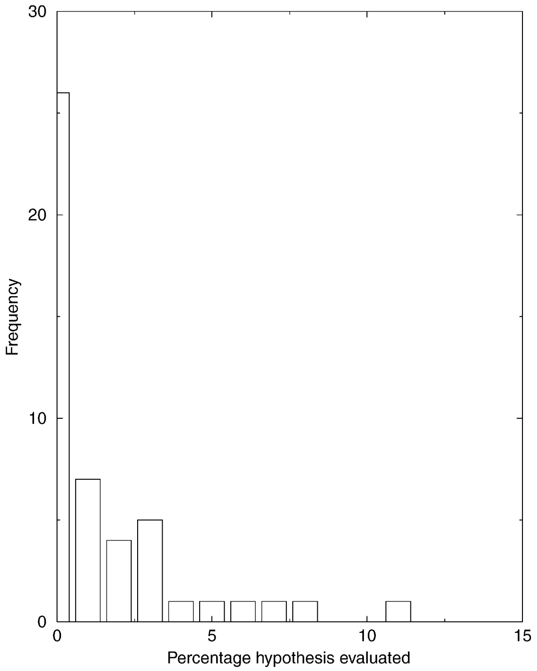

Fig. 7. Histogram of percentage hypothesis evaluated until recognition.

histogramming the samples in a coarse (32^3) RGB cube, smoothing, and normalizing.

Fifty images containing Ronaldo were gathered from the Internet, with images of varying sizes. Some images were acquired from frames of MPEG movies. The criteria used to select the images were 1) the subject in uniform should be visible in the image and 2) the images should not be close-ups (in which case, the task would be much too easy). The scale of the subject varied in height, in a range between 5 to 100 pixels. Fig. 11 shows two of the 50 images.

We used single pixels as image features. From each image, $N = 1,200$ pixels were sampled at grid points of the image. In five instances, this allowed the player to fall between gaps in the sampling and, so, sampling was quadrupled to $N = 4,800$ pixels. Although this suggests that there were as many as $4,800 \times 2$ hypotheses in any image, in reality, most of the selected pixels had RGB values that could not be produced by either color distribution. Those pixels were discarded from the count of potential hypotheses.

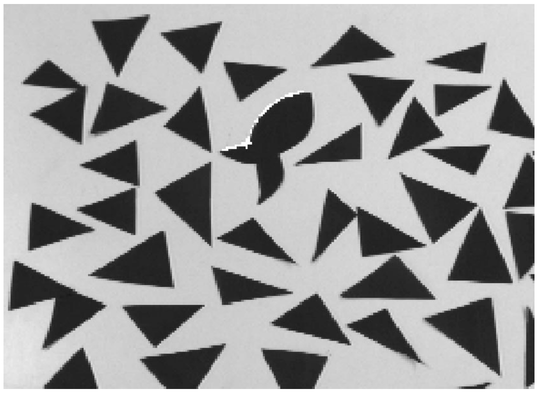

Fig. 8. Search under pop-out.

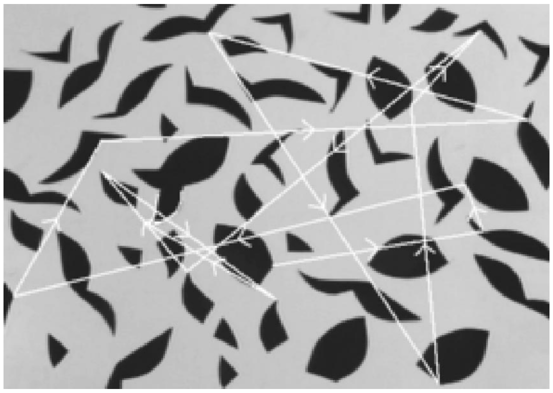

Fig. 9. Search under camouflage.

Results. For this experiment, only 0.86 percent of the possible hypotheses were evaluated by the attention algorithm, on average, before the correct hypothesis was found. Fig. 11 shows the extreme examples of pop-out and camouflage. In Fig. 11a, Ronaldo's shirt is the only yellow object in the image and it immediately pops out against a largely green background. In Fig. 11b, he is camouflaged by his teammates, who offer equally good matches to the color distributions.

6.4 Experiment 4: Color Features

As a final example, we evaluated the effectiveness of ML attention on the well-known *Where's Waldo* game—a children's book series in which the goal is to find the title character in pages filled with highly detailed illustrations. A similar attempt was also reported in [13].

The same implementation described above was used with color densities from Waldo's shirt and shorts. Because Waldo is a small figure in the image, pixels were sampled at full resolution (610 x 338). Of the 194,380 x 2 hypotheses, only 28 (0.007 percent) were examined before the algorithm suggested the target location for Waldo. The total time for the entire search process, aside from manual evaluation,

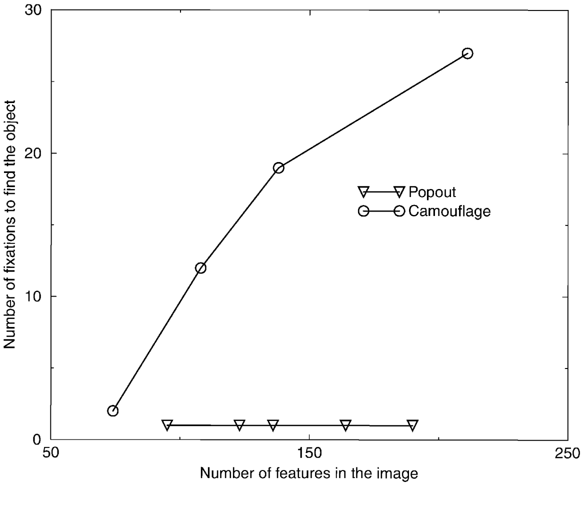

Fig. 10. Number of hypotheses evaluated vs. image features.

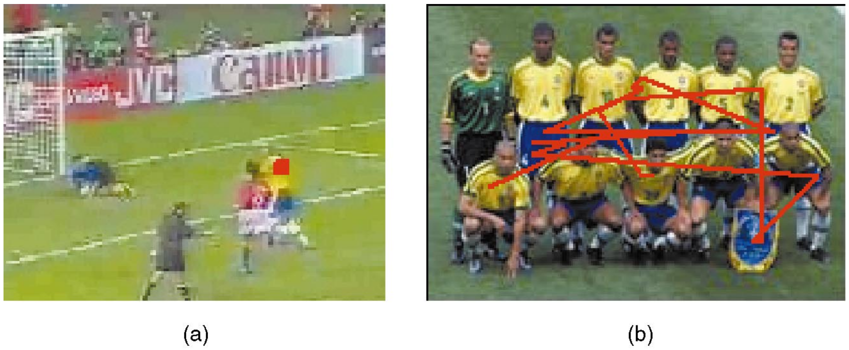
(a)
(b)

Fig. 11. Search paths for two Ronaldo images. In (a), Ronaldo was found immediately. In (b), Ronaldo is squatting at the lower left.

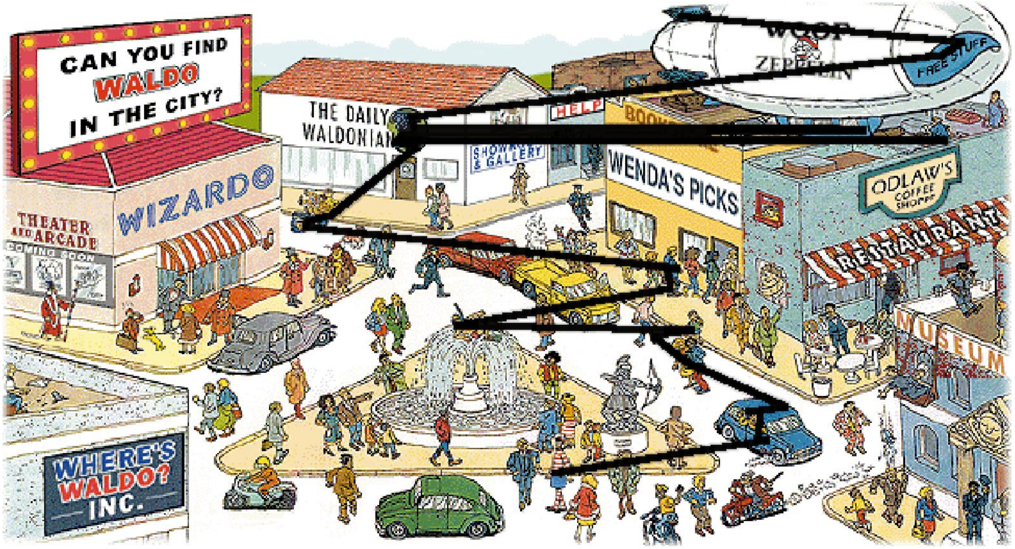

Fig. 12. Where's Waldo?

took only 0.18 seconds on a 266MHz single-processor Pentium II. In Fig. 12, the search sequence for finding Waldo is overlaid on top of the image.[1]

7 Conclusions

In this paper, we proposed a maximum-likelihood technique for directing attention. The technique uses simple features found by a fast preattentive module to direct a slower, but more accurate postattentive module. The attention mechanism recognizes that the target is made up of parts and attempts to find that pairing of target part and image feature which is most likely to come from the target in the image. The exact likelihood calculation is computationally expensive and we proposed approximations to it using r-tuples of target parts. The resulting attention strategy was shown to be adaptive for $r \geq 2$. Its choice of the part-feature pair depends on the image content. Furthermore, the attention strategy demonstrates "pop-out" and "camouflage," which are two important properties of human visual attention. In experiments with real world images, the attention strategy significantly reduces the number of hypotheses that are required to be evaluated before target is found.

Appendix

Approximation for the Distractor Distribution

In this appendix, we drive the approximation for the distractor distribution. Recall that:

N is the total number of features in the image,
M is the total number of model parts,
O is the number of model parts that are completely occluded.

Let λ be the distractor rate and V be the volume of the feature space. Let $P_i, i = 1, \cdots, M$ be the probability that part S_i is visible. The average number of parts that are visible is $\sum_i P_i$. Thus, the average number of distractors is $N - \sum_i P_i$. The distractor rate can be estimated by

$$\lambda V = N - \sum_i P_i.$$

Assuming that N is large, and using the Gaussian approximation to the Poisson distribution, the likelihood of the distractors is

[1] This particular image can be found at http://www.findwaldo.com/city/city.asp.

$$
\begin{aligned}
&p_d(N-M+O)\\
&= \frac{1}{2\pi\sqrt{\lambda V}} \exp\left\{-\frac{(N-M+O-\lambda V)^2}{2\lambda V}\right\}\\
&= \frac{1}{2\pi\sqrt{N-\sum_i P_i}} \exp\left\{-\frac{(N-M+O-N+\sum_i P_i)^2}{2N-\sum_i P_i}\right\}\\
&\simeq \frac{1}{2\pi\sqrt{N-\sum_i P_i}} \exp\left\{-\frac{M^2}{2N}\left(\frac{O+\sum_i P_i}{M}-1\right)^2\right\}\\
&\qquad \cdots \text{since } N >> \sum_i P_i\\
&= \frac{1}{2\pi\sqrt{N-\sum_i P_i}} \exp\left\{-\frac{M^2}{2N}\left(1-\frac{O+\sum_i P_i}{M}\right)^2\right\}\\
&\simeq \frac{1}{2\pi\sqrt{N-\sum_i P_i}} \exp\left\{-\frac{M^2}{2N}\left(1-2\frac{O+\sum_i P_i}{M}\right)\right\}\\
&\qquad \cdots \text{assuming } O+\sum_i P_i < M\\
&= \mathbf{C}\prod_{j=1}^{O} \exp\left\{\frac{M}{N}\right\},
\end{aligned}
$$

where C is the term that is independent of O.

Acknowledgments

We greatly benefited from discussions with Professors Drew McDermott, Anand Rangarajan, and Lisa Berlinger of Yale University.

References

[1] N. Ayache and O. Faugeras, "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 1, pp. 44-54, Jan. 1986.

[2] J.J. Clark and N.J. Ferrier, "Modal Control of an Attentive Vision System," *Proc. Second Int'l Conf. Computer Vision,* 1988.

[3] S.M. Culhane and J.K. Tsotsos, "An Attentional Prototype for Early Vision," *Proc. Second European Conf. Computer Vision,* pp. 512-562, 1992.

[4] J. Duncan and G.W. Humphreys, "Visual Search and Stimulus Similarity," *Psychological Rev.,* vol. 96, pp. 433-458, 1989.

[5] F. Ennesser and G. Medioni, "Finding Waldo, or Focus of Attention Using Local Color Information," *Proc. Computer Vision and Pattern Recognition,* pp. 711-712, 1993.

[6] W.E.L. Grimson and T. Lozano-Perez, "Localizing Overlapping Parts by Searching the Interpretation Tree," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 4, pp. 469-482, Apr. 1987.

[7] W.E.L. Grimson, *Object Recognition by Computer.* MIT Press, 1990.

[8] F.A. Haight, *Handbook of the Poisson Distribution.* John Wiley & Sons, 1967.

[9] A.H.C. van der Heijden, *Selective Attention in Vision* Routledge, 1992.

[10] G.W. Humphreys and H.J. Müller, "SEarch via Recursive Rejection (SERR): A Connectionist Model of Visual Search," *Cognitive Psychology* vol. 25, 1993.

[11] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1254-1259, Feb. 1998.

[12] T. Lindeberg, "Detecting Salient Blob-Like Image Structures," *Int'l J. Computer Vision,* vol. 11, no. 3, 1993.

[13] G. Medioni and F. Ennesser, "Finding Waldo, or Focus of Attention Using Local Color Information," *Computer Vision and Pattern Recognition,* 1993.

[14] U. Neisser, "The Processes of Vision," *Scientific Am.,* vol. 219, no. 3, 1968.

[15] K. Pahlavan and J. Eklundh, "A Head-Eye System—Analysis and Design," *CVGIP: Image Understanding,* vol. 56, no. 1, July 1992.

[16] R.D. Rimey and C. Brown, "Where to Look Next Using a Bayes Net," *Proc. European Conf. Computer Vision,* 1992.

[17] T.F. Syeda-Mahmood, "Data and Model-Driven Selection Using Closely-Spaced Parallel-Line Groups," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 881-885, 1994.

[18] T.F. Syeda-Mahmood, "Data and Model-Driven Selection Using Color Regions," *Proc. European Conf. Computer Vision,* pp. 321-327, 1992.

[19] T.F. Syeda-Mahmood, "Model-Driven Selection Using Texture," *Proc. British Machine Vision Conf.,* pp. 65-74, 1993.

[20] H.D. Tagare and J.G. Wang, "A Bayesian Strategy for Direction Attention During Object Recognition," Technical Report 96-03, Yale Univ., 1996.

[21] K. Toyama and G. Hager, "Incremental Focus of Attention for Robust Visual Tracking," *Proc. Computer Vision and Pattern Recognition,* pp. 189-195, 1996.

[22] A. Treisman, "Features and Objects in Visual Processing," *Scientific Am.,* Nov. 1986.

[23] J.K. Tsotsos, "Analyzing Vision at the Complexity Level," *Behavioral and Brain Sciences,* vol. 13, no. 3, pp. 423-469, 1990.

[24] J.K. Tsotsos, S.M. Culhane, and W.Y.K. Wai, "Modeling Visual Attention in Selective Tuning," *Artificial Intelligence,* to appear.

[25] W.Y.K. Wai and J.K. Tsotsos, "Directing Attention to Onset and Offset of Image Events for Eye-Head Movement Control," *Proc. IEEE Workshop Visual Behavior,* June 1994.

[26] J.L. Turney, T.N. Mudge, and R.A. Volz, "Recognizing Partially Occluded Parts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 4, pp. 410-421, Apr. 1985

[27] L.E. Wixson and D.H. Ballard, "Using Intermediate Objects to Improve the Efficiency of Visual Search," *Int'l J. Computer Vision,* vol. 12, 1994.

[28] J.M. Wolfe, K.R. Cave, and S.L. Franzel, "Guided Search: An Alternative to the Feature Integration Model for Visual Search," *J. Experiemental Psychology: Human Perception and Performance,* vol. 15, 1989.

Hemant D. Tagare (S'86-M'90) received the BTech degree from the Indian Institute of Technology, Bombay, in 1981, the MS degree from Rice University, Houston, Texas, in 1983, and the PhD degree from Rice University, Houston, Texas, in 1990, all in electrical engineering. Dr. Tagare is an associate professor with the Department of Diagnostic Radiology and the Department of Electrical Engineering at Yale University. His current research interests are computer vision for medicine and robotics. He conducts research on segmentation, nonrigid motion, attention in object recognition, and place recognition for mobile robots. He is a member of the IEEE.

Kentaro Toyama received the AB degree in physics from Harvard University in 1991 and the PhD degree in computer science from Yale University in 1998. Since late 1997, he has been a researcher at Microsoft Research in Redmond, Washington. His primary research interest is in vision-based tracking, with applications to human-computer interfaces, video teleconferencing, graphical avatar animation, and video understanding. For more information about his research, please see http://research.microsoft.com/toyama.

Jonathan G. Wang received the MS degree in computer science from Yale University in 1996. During that time, he worked at the Yale Vision and Robotics Lab on several projects on tracking and navigation. He is now a vice president with Credit Suisse First Boston, an investment bank in New York. He leads an e-commerce team for fixed income trading.