Vision Research 50 (2010) 1338–1352

Contents lists available at ScienceDirect

Vision Research

journal homepage: www.elsevier.com/locate/visres

A Bayesian model for efficient visual search and recognition

Lior Elazary [a,*], Laurent Itti [b]

[a] Department of Computer Science, University of Southern California, Los Angeles, CA 90089-2520, USA
[b] Department of Computer Science and Neuroscience Graduate Program, University of Southern California, Los Angeles, CA 90089-2520, USA

ARTICLE INFO

Article history:
Received 24 July 2009
Received in revised form 13 November 2009

Keywords:
Recognition
Search
Attention
Feature
Scene analysis

ABSTRACT

Humans employ interacting bottom-up and top-down processes to significantly speed up search and recognition of particular targets. We describe a new model of attention guidance for efficient and scalable first-stage search and recognition with many objects (117,174 images of 1147 objects were tested, and 40 satellite images). Performance for recognition is on par or better than SIFT and HMAX, while being, respectively, 1500 and 279 times faster. The model is also used for top-down guided search, finding a desired object in a 5×5 search array within four attempts, and improving performance for finding houses in satellite images.

1. Introduction

Attempting to search for and recognize particular known objects in a scene can be extremely complex when one has to consider all possible views an object can take. Humans employ attention to try to limit the amount of information that needs to be processed in order to speed up search and recognition (we rarely look at the sky when searching for our car). Previous research has shown that visual search tasks can be performed faster when one knows the exact target in visual space, as opposed to only a semantic description of the target (Wolfe, 1998). Therefore, humans use cues from the target image to help facilitate search. One can also consider implementing attention in the feature domain when searching through a large dataset for a particular object. For example, if we wish to search for a green bottle, we could bias the visual system so that green vertical edges would be perceived faster than other features (since bottles are often upright). This would allow us to focus a more complex recognition onto only the locations in the search scene that contain green vertical edges, which would speed up the search significantly. Likewise, during recognition, that green vertical edge may be useful to quickly narrow down onto a smaller set of possible recognition candidates. The use of various features in this manner can help sift through very large object datasets when attempting to recognize objects (consider the large number of objects that an adult human can identify). Lastly, it has been shown by Tsotsos (1991) that knowing the features of a target reduces the complexity of visual search from NP complete to linear. These findings suggest that humans employ various heuristics to improve the tractability of performing search and recognition. In this paper, we develop a model which explores the use of biologically plausible attentional heuristics to speed up search and recognition.

It is well known that the search and recognition behavior in humans can be explained through the combination of bottom-up information from the incoming visual scene (Itti & Koch, 2001; Theeuwes, 1995), and of top-down information from the visual knowledge of the target and the scene (Moran & Desimone, 1985; Motter, 1994; Treue & Trujillo, 1999; Wolfe et al., 2004; Krummenacher, Muller, Reimann, & Heller, 2001; Theeuwes, 1994; Hayhoe & Ballard, 2005). However, the exact interaction between the two processes still remains elusive, which has made it difficult to develop machine vision systems exploiting both bottom-up and top-down information.

There have been at least three major theories on mechanisms of integration between bottom-up and top-down vision occurring in the visual cortex. The first is Feature Integration Theory (Treisman & Gelade, 1980; Treisman & Sato, 1990), in which several low-level visual features are processed over the entire visual field in separate neuronal maps (called feature maps), and then combined to form a master map that guides attention. If the target can be defined by a set of primitive feature maps (e.g., it has a distinct color, orientation, intensity), then these maps can be biased using such top-down information to elicit the target location. However, if the target is defined only by some conjunctions of these primitive feature maps (e.g., a unique combination of color and orientation), then a serial search is required to find the target, since a unique signature of the target cannot be obtained from the separate feature maps alone. In

* Corresponding author.
E-mail addresses: elazary@usc.edu (L. Elazary), itti@usc.edu (L. Itti).

0042-6989/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.visres.2010.01.002

contrast, the Guided Search method proposed by Wolfe (1994) creates a master activation map where top-down knowledge is used to weigh the relative contributions of bottom-up feature maps to emphasize both features (e.g., a red color) and locations (e.g., the top-left corner of the image) likely to characterize the target. The model then uses the combination of these weighted maps to shift attention towards the most promising locations. Lastly, the Biased Competition Model proposed by Desimone and Duncan (1995) involves competition between visual stimuli at each stage of processing, which is influenced by top-down modulation. In this model, attention biases the response of a local feature detector when two stimuli are simultaneously exciting it (i.e., are presented within the receptive field of the same visual neuron). The response is biased in the direction of the attended feature in a different location. In all these models, choosing the correct feature maps to use in visual search, as well as deciding how exactly to influence these maps with top-down information, is crucial to search performance.

Previous models such as Feature Integration Theory (Treisman & Gelade, 1980; Treisman & Sato, 1990), Guided Search (Wolfe, 1994), Biased Competition Model (Desimone & Duncan, 1995) and Optimal Gains (Navalpakkam & Itti, 2006a, 2007) have largely concentrated on biasing the feature maps in a global way to facilitate efficient search (Fig. 1). For example, changing gains (or weights) over whole maps has been proposed and implemented in Wolfe (1994), Navalpakkam and Itti (2006a) and Treue and Trujillo (1999). However, simply setting feature gains globally may not always accelerate search for a target object, especially for maps that code for features shared by the target and many distractors. Furthermore, previous models have concentrated on determining the values of these gains from the objects so as to guide search towards them, but most have not shown how they can be used for object recognition. In this work a common representational framework is used for learning how to bias towards desired targets and for recognizing these targets when they are found. Thus allowing the same top-down signals or parameters used for attention biasing, to also be used for recognition.

One of the previous proposals to compute the gain or weight of particular feature maps is to base the values on the signal to noise ratio, defined as the ratio of a detector's response to the target relative to a distractor. Namely, this approach proposed that the relative weights of feature maps should be modulated top-down in proportion to each map's ability to distinguish the target from the distractors (Navalpakkam & Itti, 2006a; Navalpakkam & Itti, 2007). One shortcoming of such an approach is that, if the detectors in a given feature map respond to both the target and the distractors equally, then no change in gain will take place (Fig. 2a), which would not contribute to improvement of search speed. Moreover, if a feature detector responds more strongly to a distractor object than to the target, a reduction in gain of this map would occur, which could end up turning off this map completely. As a result, only the feature maps that can uniquely distinguish the object being searched for are amplified. Nonetheless, if a target object contains a weak red feature among strong red distractors, the weak red signal could in principle be used to find the object by guiding attention towards locations where feature detectors report low red values. Even if the feature maps are divided into sub-bands with finer granularity (Fig. 2a and b) as proposed in Navalpakkam and Itti (2006b), one can always design search arrays in which one band can code for both the target and distractors, leading to a failed discrimination.

There have also been many contributions to object recognition and search in the computer vision literature. These contributions often concentrate on two aspects of the problem: developing methods to extract features from images, and creating algorithms to classify these features. Some of the research has also independently been focused on searching for objects once particular features have been learned. For example, simple template matching (Gonzalez & Wintz, 1987; Horn, 1986; Pratt, 1991; MacLean & Tsotsos, 2008) or back-projection approaches (Bradski, 1998; Comaniciu & Meer, 1997) use some knowledge (a template or histogram) to check every possible location in the image for a good match. These techniques often fail when the object's pose or

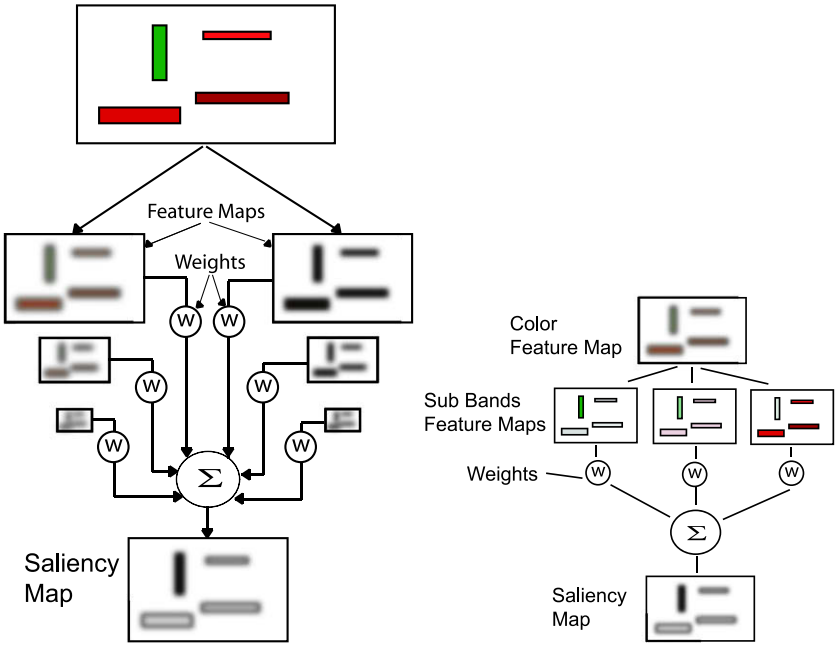

Fig. 1. Example search in previous models such as Feature Integration Theory (Treisman & Gelade, 1980; Treisman & Sato, 1990), Guided Search (Wolfe, 1994), Biased Competition Model (Desimone & Duncan, 1995) and Optimal Gains (Navalpakkam & Itti, 2006a, 2007). Left image shows the basic scheme of computing a saliency map from the weighted sum of various feature maps with varying scales (intensity, color, orientation, etc.). Biasing the saliency map towards a particular feature or scale can be achieved by changing the relative weights (w) between the feature maps. Right image shows how greater granularity in biasing can be accomplished by splitting a particular feature map into multiple sub-bands. Ultimately, both models fail to provide fine granularity in biasing for specific features (see text for explanation).

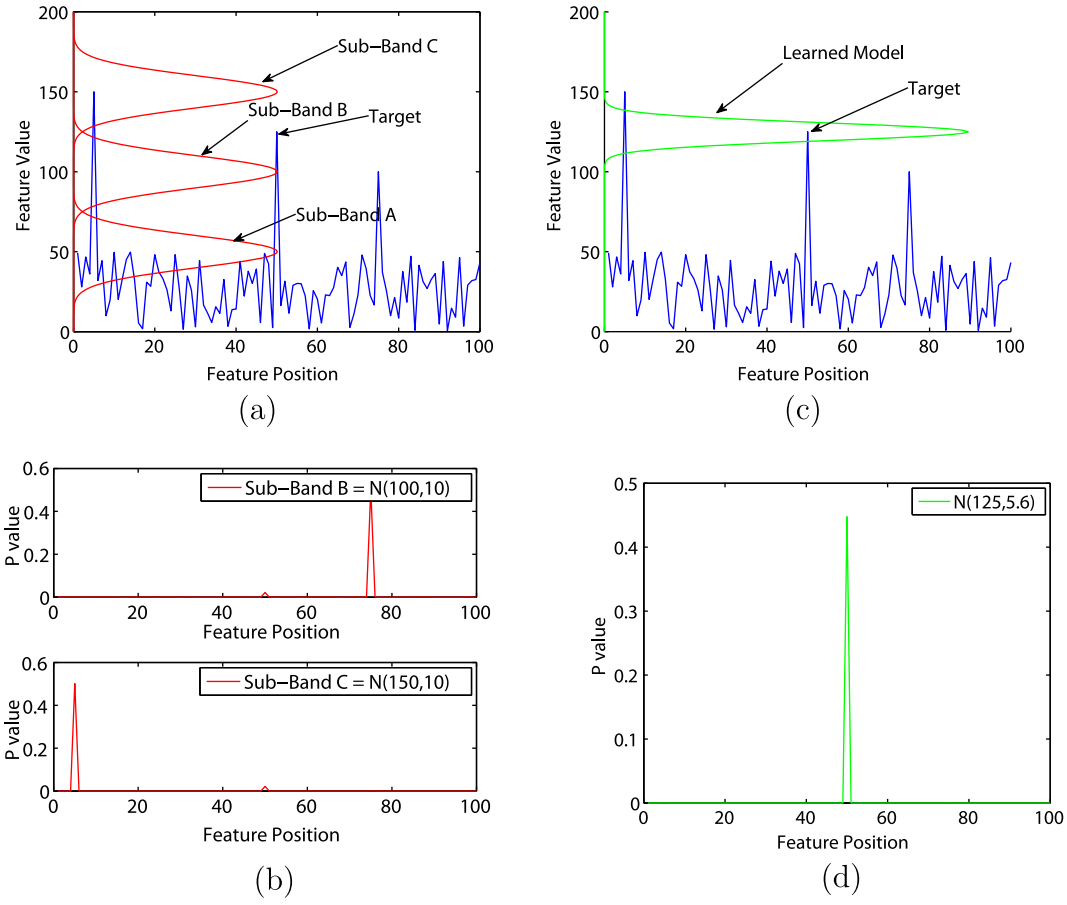

Fig. 2. An example of biasing using feature bands (a,b) and a likelihood model (c,d). In both cases the target (at spatial position 50 in a 1D slice of a feature map) has a feature value of 125. (a) Shows how three sub-bands with mean feature responses at 50, 100, 150 and standard deviation of 10 will split the feature space. (b) Shows the ambiguity in the response of sub-bands B and C when searching for the target, whereby each sub-band responds more vigorously to a distractor than to the target (sub-band A does not respond at all to the target and is not shown). As a result, changing the weight of any one of them will not yield a higher response for the target. (c) Shows how knowing the model of the target can give the granularity needed to find the target, while (d) shows the response from the learned model.

illumination is changed. To speed up search, an attentional framework proposed by Bonaiuto and Itti (2005) uses a bottom-up saliency map to rapidly eliminate locations in scenes which are unlikely to contain interesting objects. Although they reported faster results in their searches, the system lacked a method for exploiting top-down knowledge about the search target's features. Obdrzalek and Matas (2005) have also proposed a method which helps speed up the classification stage by organizing the classifier into a binary tree to achieve a log(N) time complexity. Tagare, Toyama, and Wang (2001) proposed a model in which an attentional strategy was used to reduce overall computations by performing fast but approximate image measurements. However, their computations involved finding parts of objects and determining their relationships in an approximate manner. In contrast, the contribution of the present paper is to provide a good feature set which can be quickly classified with a simple classifier, as well as the ability to use these feature sets to create a biased saliency map in order to quickly find the object in the scene regardless of pose. The methods described above can then be used to perform a more thorough evaluation of objects deemed by our system to be highly probable candidates, after these candidates have been selected in a first quick pass by our algorithm.

In this paper we draw inspiration from both the computer vision literature and models of the visual cortex and present a method based on a Bayesian framework to account for search and recognition in a probabilistic manner. In particular, a new model of combined attention and recognition is developed with dual emphasis. First, top-down biasing towards desired features should be readily available and, if possible, stronger than modulating the relative gains of different visual features guiding search, as explored in the past (Wolfe, 1994; Navalpakkam & Itti, 2007). Second, a common representational framework should be developed that can be used both for biasing towards desired targets as well as for speeding up recognition when these targets are found. We name our algorithm SalBayes which denotes our system's marriage of both saliency and Bayesian modeling.

From a biological aspect, this paper aims to develop a new approach which considers profiles of detectors that are more likely to respond to the target by shaping their tuning curve towards the target individually. In particular, we consider a Bayesian framework that uses the prior knowledge of the objects to help shape the response of the detector profile in a dynamic manner. This approach achieves greater granularity in the discrimination ability of the search without the added overhead and limitations of multiple sub-bands. Additionally, the same information learned during recognition is used to guide attention. This is achieved by learning the likelihood probability density functions (PDFs) of salient features of various objects and then using these likelihoods to compute a probable location of objects during a search task.

The result of this work is a single computationally efficient system which provides dual use. When given a location in an image, the system will output a sorted list of objects and the associated probabilities to the type of those object that can be found at the given location. Alternatively, when given a description of an object,

the system will produce a sorted list of locations and associated probabilities that the given object can be found at a particular location. From these results, other more comprehensive models (which would presumably be slower) can operate on these lists to yield robust object recognition and search. Hence, we address the problem of prioritizing search and recognition, narrowing down from long and unordered to shorter and ordered lists of candidates, rather than completely solving and outputting a single recognized object label at a single location. We show how this is achieved by learning the visual features of an object, which is used for recognition as well as for efficient top-down-guided search. In testing against large standard databases (Amsterdam Library of Object Images (ALOI) (Geusebroek, Burghouts, & Smeulders, 2005), Columbia Object Image Library (COIL) (Nene, Nayar, & Murase, 1996), and SOIL-47 (Burianek, Ahmadyfard, & Kittle, 2001)), we find that this approach delivers robust machine vision performance, comparable and much faster than other more sophisticated, computationally intensive, and state-of-the-art machine vision systems (HMAX, SIFT) for recognition, while additionally providing a common framework for search and recognition.

In the following section we describe the model and its components. We start with the simple problem of object classification, and of defining a representation that can be learned from example views of objects. We then explore how this representation can also be used to provide efficient visual search for the learned objects. Section 3 describes the testing methodologies, datasets and results, while Section 4 provides discussion of the model and results.

2. Methods

The model proposed in this paper draws its inspiration from Bayesian theory as well as from the bottom-up attention model proposed by Itti et al. (Itti, Koch, & Niebur, 1998; Itti & Koch, 2000). By learning the statistical variations in features of various objects, the model is able to perform an efficient visual search for a given target object, as well as classify target and distractor objects. At its core, the model learns the probability of an object's visual appearance having a range of values within a particular feature map. In a visual search task, the model influences the various feature maps by computing the probability of a given target object for each detector within a feature map. As a result, locations in the maps with the highest probability will be searched first, as they indicate likely positions for the target object. Both the prior and likelihood probabilities can be learned from training views of the object and the context. As we will see, a chief advantage of this approach is in its simplicity and speed, which make it an ideal candidate for a front-end system that quickly narrows search down to a few likely candidates which can then be investigated in more detail by more sophisticated and time-consuming recognition algorithms.

2.1. Object representation

To uniquely describe the appearance of an object, a number of feature maps are computed from the biologically inspired bottom-up saliency model proposed by Itti et al. (Itti et al., 1998; Itti & Koch, 2000). The saliency map represents statistically unique locations in an image after being decomposed into different feature maps at several spatial scales. That is, the saliency map attempts to detect anomalies, or outliers in the image within various feature spaces. In this paper the feature map domains consist of intensity, color opponency (red–green, blue–yellow) and four orientations (0°, 45°, 90°, 135°). These particular feature maps were selected based on the implementation proposed by Itti et al. (1998) which derived its inspiration from a review of which elementary visual features contribute to visual saliency in natural scenes (Wolfe et al., 2004). In the absence of top-down modulations a normalization operator, $N(.)$, within each feature map weighs the values of detectors in a data-driven fashion based on their uniqueness in that map. That is, the more different the response of a given detector is from its neighbors and globally, the higher the weight assigned to that detector's output. This normalization operator can also be thought of as providing spatial competition between neighboring pixels. The normalizing operator is computed as follows (Itti et al., 1998):

1. Normalize the values in the map to a fixed range $[0\ldots,M]$, in order to eliminate modality-dependent amplitude differences;
2. Find the location of the maps global maximum M and compute the average $\bar{m}$ of all its other local maxima; and
3. Globally multiply the map by $(M-\bar{m})^2$

The 42 feature maps (seven features at six spatial scales) are then combined into a saliency map, which indicates the saliency of each location in the image. Implementation details of this model have been described previously (Itti et al., 1998; Itti & Koch, 2000) and the algorithm is freely distributed in source code at http://iLab.usc.edu/toolkit/.

To characterize the target, the most salient features from each of the 42 feature maps are sampled within a given fovea size (or patch size) centered on the object. Note that this location does not need to be the center of the object, nor does the object need to be segmented. The only requirement is that the object should overlap with the fovea location. The spatial competition will help provide a consistent location from which to sample when the object undergoes various transformations (illumination direction, rotation, etc.). Selecting the most salient location to learn from also helps in searching for the object. For example, if we know that we are looking for a red dot on the object, then it's worth searching for a red dot.

The motivation behind sampling from the most salient location within each submap around the object is to select features that would uniquely describe the object, but would still remain invariant to transformations in illumination, rotation, translation, etc. This can then provide a very efficient search mechanism when attempting to narrow down possible objects during recognition. The argument follows that a salient location in an object would remain invariant to transformations since it is very unique to the object. For example, the model not only learns that the object has a particular strong color value, but also that it has a particular strong intensity, and particular orientations. Therefore, not only the conjunctions of various feature maps can yield a position that is highly salient, but also feature values within each feature map at these strong locations.

The method of only selecting particular key locations to describe objects and scenes, rather than considering the entire pixel array, has also been successfully used by the SIFT algorithm (Lowe, 2004) and has been studied by Mikolajczyk, Leibe, and Schiele (2005). However, this paper uses the saliency map described above which is a much more elaborate method of determining the keypoint locations in order to provide a more robust feature set for recognition. Note that only the single most salient location in each feature map is used to build the descriptor vector. This results in very quick recognition rates, since adding more locations would require a more complex model to account for spatial locations within them. In particular, the goal of the model is to code probable locations and or hypotheses of particular objects, but not determine them specifically. Therefore, we would want to use as few features with a few complexities in order to speed up initial recognition and search. Other, more complex models (which would require more time to compute), would then be used on these locations in order to specifically determine the object.

Nevertheless, since the features are sampled from multiple scales, some spatial information is encoded in the feature vector but is not tightly localized. This is a result of sampling from the various scale-space pyramids. Consequently, features extracted from a single salient corner of a rectangle will yield a different signature (vector of features) than a signature obtained from a square. This is due to the fact that a rectangle will occupy a different number of cells within the image, and thus will show up in a different pyramid level. Additionally, a more complex model (with multiple features and locations) can be considered but would result in less efficient recognition or search (especially when the spatial distributions of the features are included). For a similar approach to recognition with a more complex model (see Shokoufandeh, Marsic, & Dickinson, 1998).

During training, the object model descriptor is built by computing the likelihood probability distributions of the 42 features resulting from each feature map. This PDF is modeled using a Gaussian distribution for each individual feature type, where both the mean and variance are learned. That is, the algorithm learns 42 separate univariate Gaussian distribution for each object. The choice of this distribution is due to the simplicity and efficiency in obtaining the parameters mean (μ) and variance (σ^2) in an on-line method from training images. Additionally, these likelihoods are used later for searching for the object. However, other distributions can be used such as super-Gaussians, mixtures of Gaussians, particle filters, or discrete histograms (Scott, 1992).

Given a region of interest patch q with N pixels from a particular location (this location will correspond to the image being trained with) from within a given feature map (from the 42 feature maps computed above), the spatial competition method $N(.)$ (the non-linear normalization method described above) is applied to this patch to form a new set of patch values q'. A feature vector F is then built using the value of q from the location at which q' has a maximum response. This value then forms the jth component of the feature vector F, and is denoted F_j. In other words we select the center-surround feature that has the highest value in the spatial competition layer (the most unique feature in that map).

$$F_j = q[argmax(q'_i)_{i=1,\ldots,N}] \ \forall j \in F \tag{1}$$

where i represents the pixel position within the patch, F_j is the particular feature value from feature map j and F is the set of feature maps.

The Normal distribution is then used to estimate the likelihood, $p(F_j|\theta_j)$, of observing feature F_j given a particular object class parameter for this feature θ_j. For example, if j is the index of the vertical Gabor detector channel, then F_j would represent the response of that channel at it's most 'unique' location, as determined by $N(.)$. θ_j would then represent the learned mean and variance of the vertical Gabor responses for object θ.

The final model (θ) is then a set of n parameters (θ_j), each composed of a mean (μ) and a variance (σ^2) for each individual feature map. This gives the ability to simply compute the model parameters (θ) mean (μ) and variance (σ^2) from the training views of the object within each feature map, and to use a Gaussian distribution to estimate the likelihood.

$$p(F_j|\theta_j) \propto N(F_j;\mu_j;\sigma_j) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-(F_j-\mu_j)^2/2\sigma_j^2} \tag{2}$$

When learning from only a single view, the standard deviation (σ) is initially set to a fixed value of 0.001, which was chosen arbitrarily (this number should be small so that the particular feature detector will provide some discrimination). This gives the classifier a rough estimate of the classification of the object with only one training view of the object, while fully computing the variance requires more than one training view.

2.2. Object classification

To classify particular features obtained from the feature maps, a naive Bayesian network is used. The choice of a naive Bayesian network in the model was made to reduce the amount of computations necessary for classification, as this type of network assumes statistical independence between feature values. Since some of the features are derived from different scales in the image, our features are actually guaranteed to be statistically dependent. However, it has been shown that even if the features are statistically dependent upon one another, computing the full network often only increases accuracy by a small amount, whereas the computations necessary to achieve this small improvement are large (Rish, 2001). As further evidenced in Vasconcelos and Vasconcelos (2009), for image classification, modeling the joint distribution between pairwise features provides often only a marginal performance boost.

Once a set of features (F) is collected from a given salient location within the feature maps (as described above), the classification is performed using Bayes formula:

$$p(\theta_i|F) = \frac{p(F|\theta_i)p(\theta_i)}{p(F)} \tag{3}$$

To make a decision as to the type of classification assigned to an object, i can be iterated over all known objects and the object with the greatest posterior can be chosen as the best match. This method is known as *Maximum a Posteriori* (MAP). However, the goal of the model is to act as a fast front-end to slower, more accurate object recognition systems, and so we instead output a list of objects and match likelihoods sorted by the probability that each object matches the requested location. In our experiments, the prior is taken to be $1/C$, where C is the number of classes. This results in each class being equally probable to observe (uninformed prior). However, changing the prior in response to outside knowledge, could yield better classification rates if within a given scene the probability of a particular object appearing can be determined.

Since the probability of the evidence can be viewed as a normalizing constant (used to ensure that probabilities all add up to unity), it can be dropped from the equation. This is because the comparison of the posterior is between classes, and only the greatest one is selected and not its particular value (the scale of the value is insignificant). Furthermore, the assumption that features are statistically independent from one another simplifies the calculation to just multiplying the likelihoods together to come up with a decision, as opposed to calculating the full joint distribution between the features.

$$p(\theta_i|F) = \text{argmax}_i\left(p(\theta)\prod_{j=1}^{n} p(F_j|\theta_{ij})\right) \tag{4}$$

Additionally, taking the product of many probabilities, some of which may be very small, can give rise to numerical instability. As a result, an underflow often occurs in a straightforward implementation of Eq. (3) when using more than a few features. A solution to this problem is to take the log of the likelihood which will convert the probabilities from being less than one to negative numbers greater than one. This also greatly simplifies the computations in our practical implementation, as likelihood products are transformed into likelihood summations. Also, the decision to select a suitable classification is not affected, since only the maximum of these values is considered. As a result of these various techniques, Eq. (3) can be described by the following formula:

$$p(\theta_i|F) = \text{argmax}_i\left(p(\theta)\sum_{j=1}^{n} \log(p(F_j|\theta_{ij}))\right) \tag{5}$$

The enhanced version of the saliency map with the Bayesian network used for object recognition can be seen visually in Fig. 3.

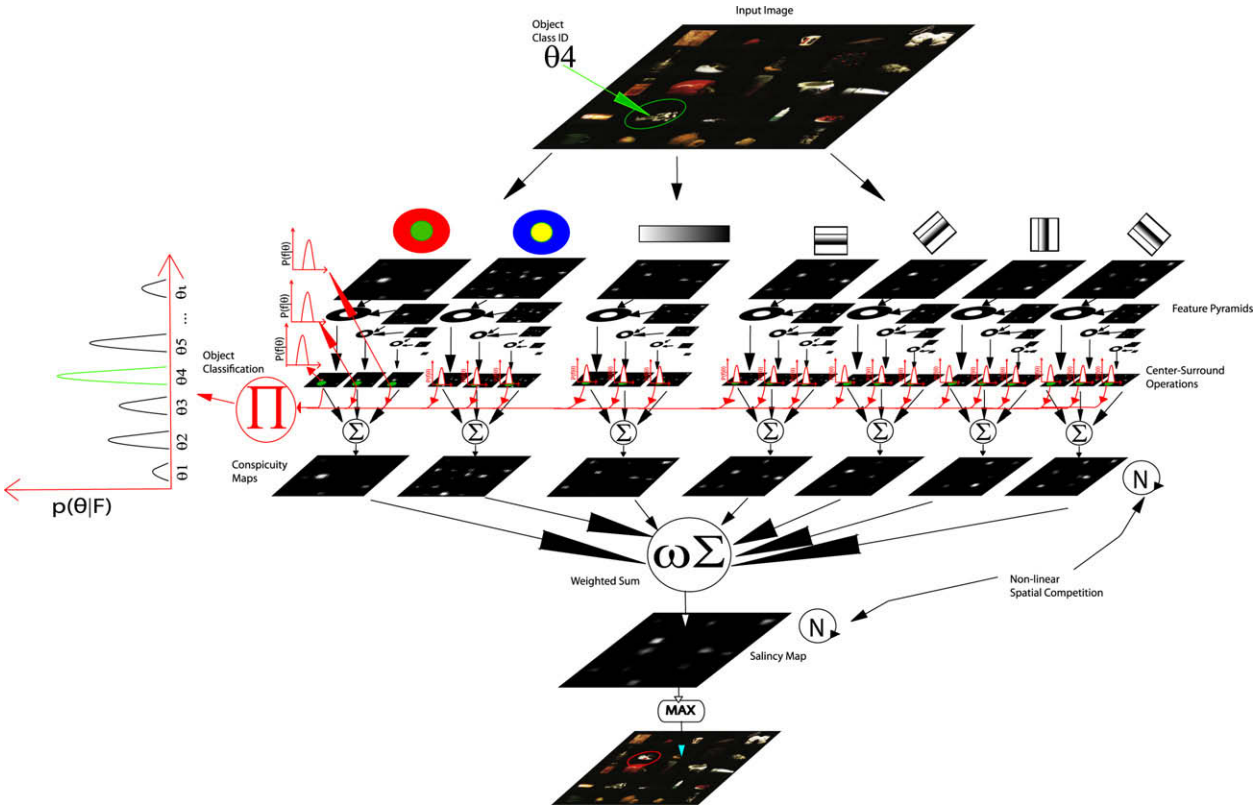

Fig. 3. The added Bayesian network to the saliency model for object recognition. Red indicates added components and data paths. The toy soldier at the input image is selected for learning/classification indicated by a green circle. The maximum feature location in each center-surround feature map is used to train or classify the Bayesian network for the selected object (indicated by the probability map on the left side). Each feature map builds a probability distribution of the most salient location in that map, shown in red. The rest of the architecture is as previously described.

2.3. Biasing learned features for efficient object search in a Bayesian framework

Once the parameters of a particular object are known, they can be used to search for the object in an efficient manner. This is accomplished by biasing the feature maps to influence the saliency map so that the object that is being searched for becomes more salient, which can result in a faster search times simply by sorting by salience. For example, if our bottom-up saliency computations considered bright locations as salient, then darker locations would often be considered last as possible targets. However, if our object was dark in color, then biasing the saliency computations to choose darker locations as salient should improve search time, which would result in the biased saliency highlighting darker locations first.

The saliency map is biased by using the knowledge of the target parameters, and applying them to the set of feature detectors that are computed. Particularly, the parameters of our target are used to look for a particular mean and a variance within a given feature map. These parameters can be thought of as an envelope limiting the feature map response. In other words, the feature map would have its activation shaped by the likelihood of the particular feature value belonging to the object. Although our system could be thought of as only considering one sub-band, that sub-band can be dynamically shaped (regarding its position along the feature spectrum, and its specificity or width), thus providing an interesting alternative to using several fixed sub-bands. The result in the feature map then gives the probability of our object being coded by a particular feature detector. The maximum location within the feature map would then give an indication of the possible target location (Fig. 2c,d). The biasing process (applying the likelihood model to the feature map) is repeated within each feature map and the combination of all the feature maps' information is used to create a saliency map where the maximum indicates the most probable location of our target. The enhanced version of the saliency map with the Bayesian network used for finding objects can be seen in Fig. 4.

The various feature maps in the saliency map are biased in the following way: First the feature maps are computed by creating an image pyramid of each feature type and taking the difference between the pyramids to form center-surround responses at various scales as proposed in the original saliency algorithm (Itti et al., 1998; Itti & Koch, 2000). There are 42 such maps labeled $F_1 \cdots F_{42}$ (four orientations, one intensity, one blue–yellow, and one red–green all at six different scales). Note that spatial competition is not computed on these feature maps and just the raw center-surround values are used. However, it is important to remember that the spatial competition was used when extracting the feature values during the training phase. From the learned parameters of a particular object θ the parameters (μ_j and σ_j) corresponding to a particular feature map F_j are used to calculate the probability of a particular detector belonging to the target $p(F_j|\theta_j)$ in feature map j. This is done across the n different feature maps. The maps are then multiplied together (instead of the sum which was used in the original model) to yield the final saliency map. Therefore, the resulting saliency map calculates the probability of a location containing a given target ($p(F|\theta)$). Again, to avoid numerical instability and to speed up computation, the log of the probability is used.

$$\log(p(F|\theta)) = \log(\prod_{j=1}^{n} p(F_j;\theta_j)) \propto \sum_{j=1}^{n} log(N(F_j;\mu_j;\sigma_j)) \quad (6)$$

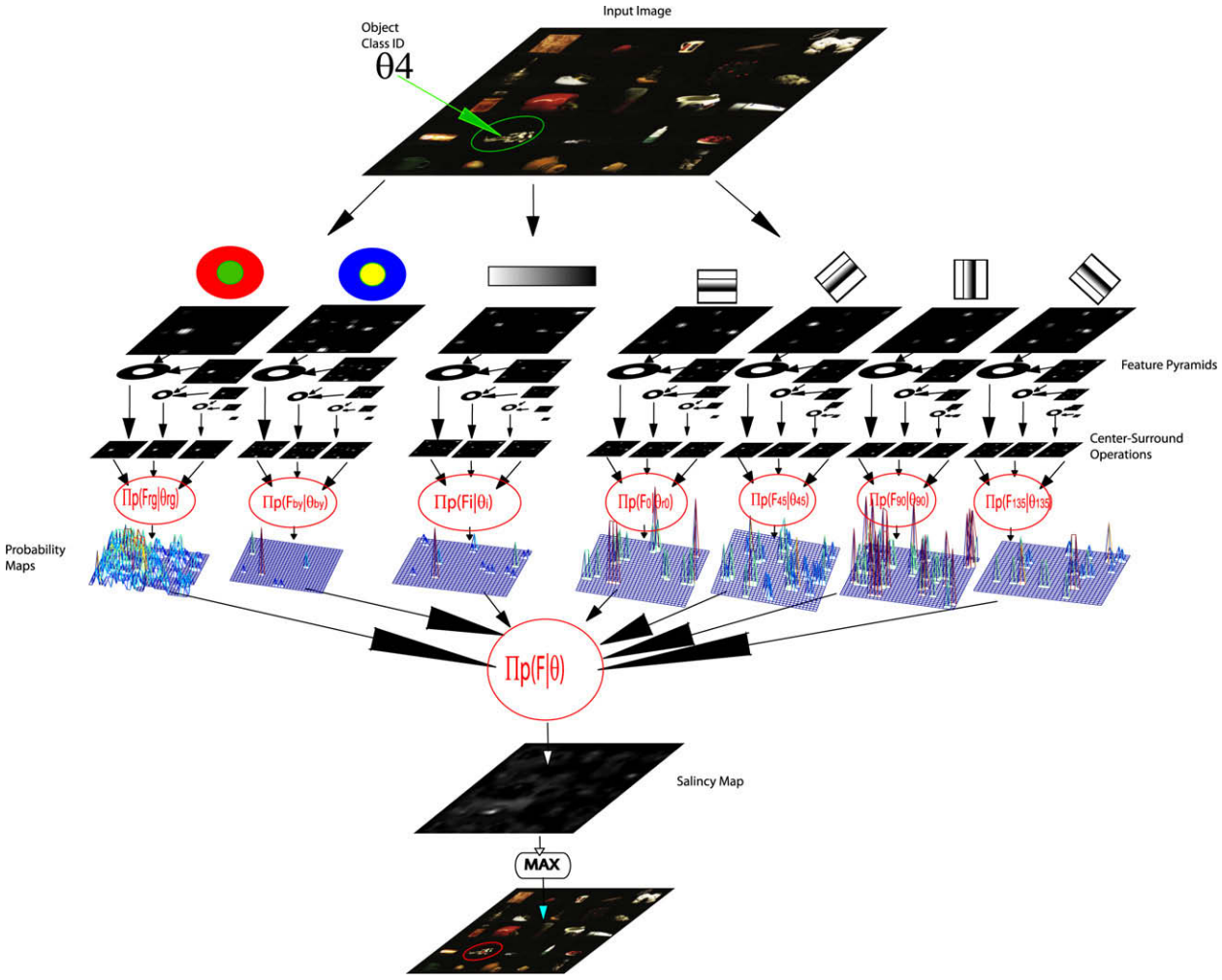

Fig. 4. The added Bayesian network to the saliency model for object recognition. Red indicates added components and data paths. The input image is passed for processing (without the selected object which is indicated in this image for clarity) by the saliency computations in the normal way. After the center-surround operations, the parameters of the object are used to find the probability of a detector indicating the position of the object in each submap (depicted as 3D graphs in the figure). All the submaps are then multiplied together to form the saliency map (note that in the implementation the multiplications are converted to additions by the used of the log operation).

N is the normal function and F is the set of all 42 features. Again, the spatial competition on the whole saliency map is not performed during object search. This is due to the nature of the spatial competition, which tends to punish high values within the same uniform region. Since that region describes the probability of the object, that location should be allowed to contribute to the overall saliency map.

Once a biased saliency map is computed, the locations with the highest locally maximal values in that map are searched first. That is, the object model is used on the locations which are local maxima in the biased saliency map. This processes in known as attention shifts. Finding local maxima is achieved by selecting the maximum value in the saliency map and applying an inhibition of return (IOR) mechanism to that location. The IOR is performed by applying a Gaussian disk mask with fixed radius to the saliency map which set all salient values underneath the mask toward zero, so that the next maximum salient location would have to be outside the disk. Implementation details of this mechanism have been described previously (Itti et al., 1998; Itti & Koch, 2000).

3. Results

The model was tested on three publicly available datasets to evaluate its performance in both object recognition and object search tasks.

3.1. Object recognition

For the object recognition task, several challenging datasets were used. These datasets included objects under many transformations including rotations and various viewpoints, illumination changes and illumination color changes. The original idea of the experiment was to use SIFT (Lowe, 2004) on top of the output of our model in order to speed up the search for object during recognition. However, during our preliminary experiments we found that using SIFT did not actually provide better recognition results than the raw output of our model. As a result, we directly compare the recognition capabilities of SalBayes against state-of-the-art object recognition methods: the SIFT (Lowe, 2004) algorithm as proposed by Lowe and the HMAX algorithm with feature learning proposed by Serre, Wolf, and Poggio (2005). These two methods were chosen due to their popularity in the machine vision and cognitive modeling community. For example, HMAX has been used to explain basic facts about the ventral visual system (Riesenhuber & Poggio, 1999) and has been used in object recognition (Serre et al., 2005, Serre, Wolf, Bileschi, Riesenhuber, & Poggio, 2007), while SIFT has been used to build 3D models of objects (Snavely, Seitz, & Szeliski, 2006), robotics navigation (Se, Lowe, & Little, 2002; Elinas & Little, 2005; Sim, Elinas, & Griffin, 2005; Barfoot, 2005) and object classification (Lowe, 1999). Due to the large amount of data, a Beowulf cluster consisting of eight dual–dual-core Opterons(tm)

(four cores per node for a total of 32 cpus) running at 2.6 GHz was used to run the algorithms in a parallel fashion. The cumulative amount of CPU time taken for the testing sets was captured to compare the efficiency of the models.

The SIFT implementation was obtained from the author's website (Lowe, 2004), but the matching keypoints software was changed slightly to provide keypoint matching against a large dataset. In particular, a k-nearest neighbor algorithm (with $k = 2$) was used to determine the object identity given a test image and an image database. An implementation of HMAX with feature learning in Matlab was obtained directly from the author's website (http://cbcl.mit.edu/software-datasets/standardmodel/index.html). However, due to the large amount of data, the software was slightly modified to compute the features for all objects under all transformations and save them to a file first. This allowed us to extract the features in parallel using the Beowulf cluster. An SVM algorithm with a RBF kernel was used for training and testing. The implementation of the SVM was obtained from Chang and Lin (2001).

We test the proposed algorithm (along with HMAX and SIFT) against three large standard databases (ALOI, COIL, SOIL-47) separately and all together. The datasets are systematically broken into training and testing sets composed of the various images in the dataset. These sets include 1 image for training and the rest for testing, 6.25% training 93.75% testing, 12.5% training 87.5% testing, 25% training 75% testing and 50% training and 50% testing. The first object recognition dataset used was the Amsterdam Library of Object Images (ALOI) (Geusebroek et al., 2005). This dataset contains photographs of 1000 objects placed on a turntable and subjected to various transformations. These transformations include 12 illumination colors, 24 illumination directions, and 72 viewpoints (each object was rotated in steps of 5°). All photographs were first scaled down to a 256×256 pixel image to speed up computations. Several splits of the entire dataset into training and testing sets were considered, from using only one instance of each transformation (three images total) for training, to using half of the dataset for training. Object recognition testing was then performed on all 1000 objects on transformations that were not in the training dataset. The next data set used was the Columbia Object Image Library (COIL) (Nene et al., 1996) which consisted of photographs of 100 objects under 72 rotated views. The 7200 color images of 128×128 pixels were obtained by placing objects in the center of a turntable that was rotated at 5° increments. Here again several splits into training and testing sets were tested, from using only the first image for training and all others for testing, to using half of the dataset for training and the other half for testing. Object recognition was then performed on all 100 objects and on views that were not in the training datasets. The last dataset used was the SOIL-47 (Burianek et al., 2001) comprising photographs of 47 household objects. The images were obtained by placing a camera on a robot arm and moving it to various positions. In addition, the objects were also subjected to two illumination conditions. We again created training sets that ranged from just a single instance of each object, to half the dataset of the various views of the object. In addition, one of the illumination conditions for each of the two illumination conditions was used for training. Testing was then performed on all objects and on views that were not in the training datasets.

The results show that under many object transformations the model was able to successfully learn the objects, classify them correctly and search for them in an efficient manner. In particular, Fig. 5 and Table 1 shows that the model was able to classify the large datasets tested on average over 88.64% correctly. As indicated in Fig. 5, the HMAX algorithm was able to achieve a 92.46% classification rate on the ALOI dataset. Although this is a slight improvement over the proposed method, it should be noted that the features computed in the HMAX algorithm are 2000 dimensions in size and take more then 46 s to compute per image, as compared to the proposed model which uses only 42 features and is 279

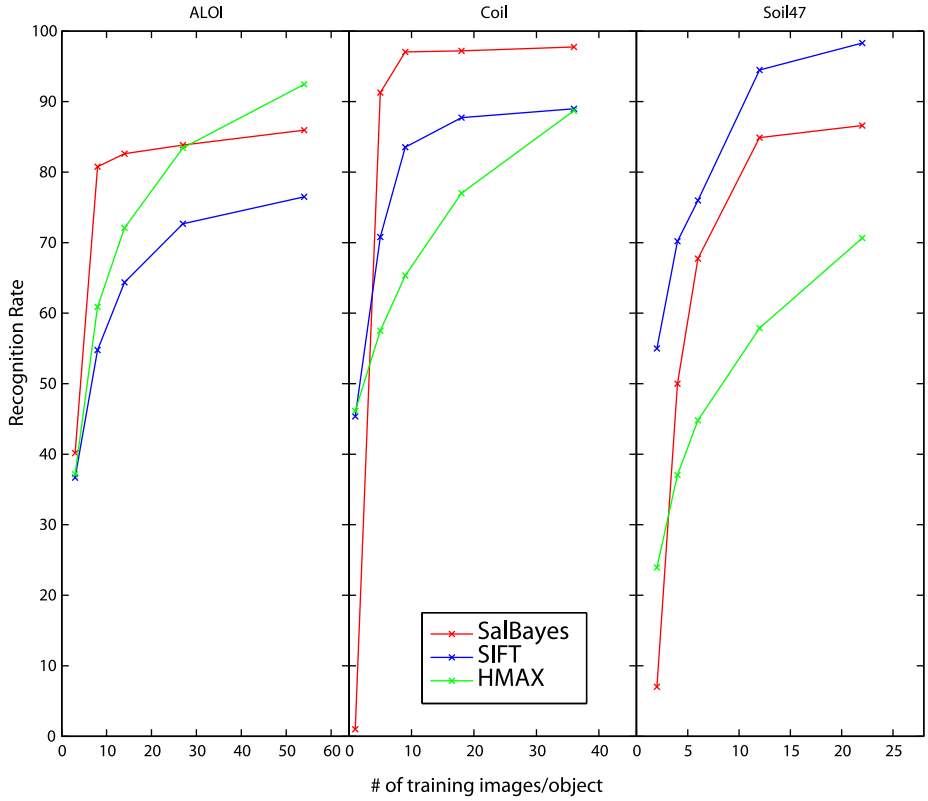

Fig. 5. Classification rates as a function of training size obtained by the proposed algorithm SalBayes, SIFT and HMax.

Table 1
Average recognition from the various datasets using 25% of the data for training. *N* represents the number of images in the testing set. The work of others have been included in this table to place the performance in context. To our knowledge, no one has before us used all of the 1000 objects in the ALOI database under all conditions.

Method	Classification rate (%)			
	ALOI *N* = 81,000	COIL *N* = 5400	SOIL47 *N* = 1410	ALL datasets *N* = 87,810
SalBayes	83.83	97.20	84.89	88.64
SIFT	72.68	87.19	94.48	84.78
HMAX	83.42	77.02	57.87	72.77
MNS (Murthy, 2007)	–	99.91	100.00[1]	–
LAF (Obdrzalek & Matas, 2002)	–	99.90	100.00[1]	–
Graph matching (Kittler & Ahmadyfard, 2001)	–	–	73.0	–
Extra trees (Maree et al., 2005)	–	99.50	–	–
Sub-windows (Geurts et al., 2004)	–	99.61	–	–
SNoW/edge (Roth et al., 2002)	–	94.13	–	–
SNoW/intensity (Roth et al., 2002)	–	92.31	–	–
Linear SVM (Roth et al., 2002)	–	91.30	–	–
NN (Roth et al., 2002)	–	87.50	–	–

Table 2
Recognition results on the ALOI dataset under the various transformations using 25% of the data for training. *N* represents the number of images in the testing set. The fifth column is the performance rate obtained when using all images (all images from A, B and C), while the sixth column represent an unweighted average of performance obtained for A, B and C (if the same number of transformations where equally likely to occur.).

Method	Classification rate (%)				
	A. Changes in illumination color only *N* = 9000	B. Changes in illumination direction only *N* = 18,000	C. Changes in rotation only *N* = 54,000	All images from A, B, and C *N* = 81,000	Unweighted average of performance for A, B and C
SalBayes	64.79	75.50	89.71	83.83	76.67
SIFT	89.41	71.47	70.95	72.68	77.28
HMAX	99.04	83.13	80.76	83.42	87.64

times faster (0.165 s per image). Moreover, the increase in performance was only achieved when training on half of the dataset, which means that the difference between a training image and a testing image is not large. As seen, the proposed algorithm, SalBayes, was able to achieve better performance with less training data at speeds which greatly surpass both HMAX and SIFT. Examining the different datasets, it can be seen that the proposed model was able to learn the object features from only a few training examples (less then five per object) and achieve good results. In particular, the COIL dataset shows that from five training examples, the model was able to correctly identify 91.28% of the test images correctly. Lastly, the test result also show that the system performs fairly well when using only gray value images (just like HMAX and SIFT). This indicates that the proposed system can still provide useful information in the absence of color information.

Because the ALOI dataset contained the most systematic transformations, further analysis was done to determine the classification rate under each type of transformation. Looking at Table 2 we can see that HMAX performs best under several transformations. In particular, it does exceptionally well on the illumination color task. On the other hand, our new model performs well in the illumination color task when considering gray value images. Additionally, the model does exceptionally well under the rotation task. This shows the model's robustness against rotation and possible other 3D transformation (as can be seen in the soil47 dataset) as a result of picking the most salient features to remember when determining the classification of an object.

Looking at the timing aspects of the models tested, it can be seen that the proposed method, SalBayes, outperforms both SIFT and HMAX by many folds. Examining the results from Fig. 6, it can be seen that for testing on half of the ALOI dataset, it took only 3.42 h for the SalBayes algorithm as opposed to 4878.3 h for SIFT and 678.55 h for HMax. On average across all the datasets the new model was more than 1500 times faster than SIFT and 279 times faster than HMAX.

3.2. Grid based object search

The visual search task was evaluated by creating a dataset which consisted of search arrays created from the ALOI images.

Figs. 7 and 8 shows an example of a scene created for the search task. The scenes were created by taking random objects from the ALOI dataset under random transformations (from all 1000 objects) and placing them in a 2×2 or a 5×5 grid pattern. A random object was then chosen as the target object and the system searched for that target. This resulted in search images of size 512x512 pixels for the 2×2 grid and 1280×1280 for the 5×5 grid (256×256 pixels per object). The parameters for the objects that were learned from training on half of the dataset as described above were used in this task. The number of "attention shifts" (inspections of individual objects) taken to reach the target object was then recorded. The inhibition of return (IOR) size was set to 30 pixels radius. This meant that only a small portion of the image would be inhibited at a time. As a result, multiple fixations per object could result if the object has strong multiple salient location, which would lead to greater number of fixations than the grid will allow.

Fig. 8 show the number of scenes vs. the number of attention shifts taken to reach the target object. The result show that during the search task, only 4.2 attended locations were required on average (with standard deviation of 5.9) to be examined in order to find the target object. About 218 fixations (290 fixations for 30 pixels IOR for the whole image minus 72 fixation for one 256 image) would be needed to systematically cover the whole image for the 2×2 and 1692 fixations for the 5×5 array. In particular, in these synthetically-generated scenes, the model was able to find the target object in fewer than five attended locations in over 76% of the scenes (average of 5×5 and 2×2 search arrays). Since the scenes could have contained any one of the 1000 objects, the ambiguity in the various scenes is large. For example, a few objects are green boxes, where the only varying feature is the size. Additionally, in

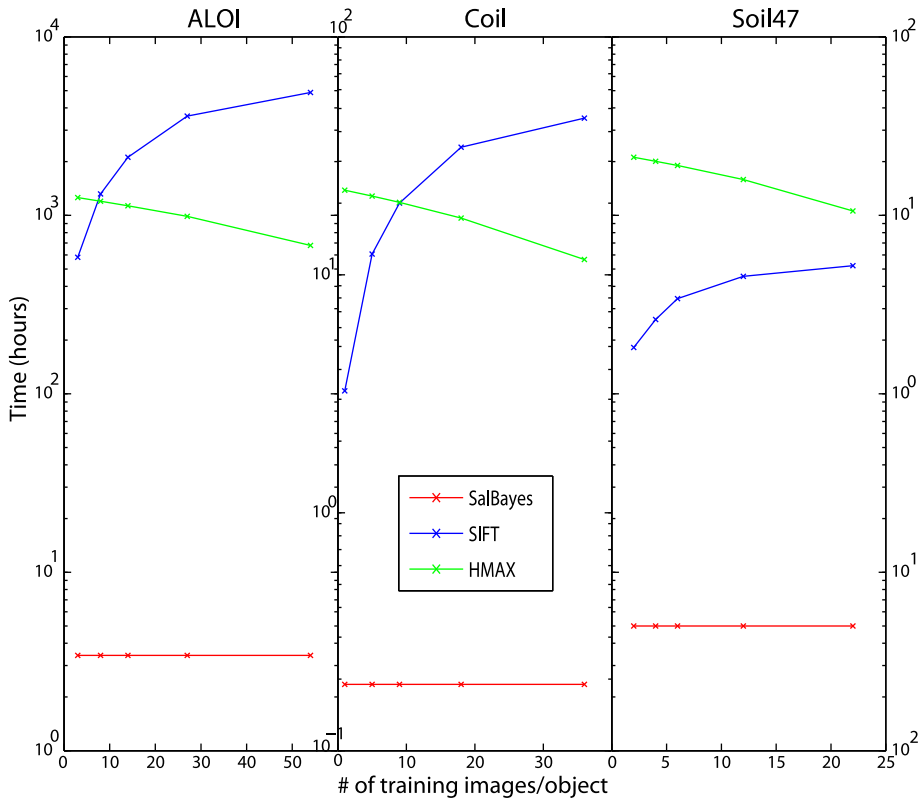

Fig. 6. Total CPU time required for testing, as a function of the fraction of each dataset that was used for training. As more images are used for training, fewer images remain for testing (hence the decrease in processing time for HMAX), but, in the case of SIFT, a larger keypoint database is built.

Fig. 7. Example 5 × 5 search scene built from the ALOI dataset. Scenes were created by taking random objects from the ALOI dataset under random transformations (from all 1000 objects) and placing them in either a 2 × 2 or a 5 × 5 grid pattern.

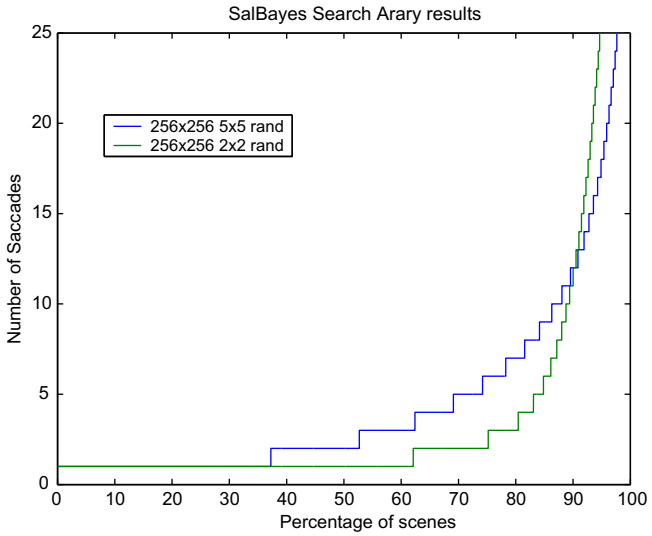

Fig. 8. Search results for the various scenes. The number of attention shifts (saccades) is plotted against the percentage of all scenes. For about 60 percent of the 2×2 scenes the object was located within the first fixation, and for about 38 percent of the 5×5 scenes the object was located within the first fixation.

some of the images, the object was never found due to a zero saliency value. Presumably, an exhaustive search would take place on the locations that were not searched.

3.3. Satellite image search

Another search task consisted of finding houses in satellite images. This task consisted of satellite images (786×786 pixels) obtained from the New Orleans region after hurricane Katrina. An example application of this type of search would be to determine the number of houses effected by a natural disaster in an autonomous manner. This can be achieved by comparing the number of houses found before and after a disaster. Since satellite images contain a lot of data, it is often difficult for humans to quickly find places of interest in these images. In this task, the model was set to find images containing houses, so that humans can determine what do to with these regions (provide food, estimate the disaster area, etc.) The system was trained with 38 instances of houses obtained from 10 such satellite images (786×786 pixels), and was tested on finding 95 houses spread out across 20 images. On an average each image contained five houses with standard deviation of 1.94 while occupying about 50×50 pixels. All images were hand labeled and a house was considered found if it was within a 30 pixel radius region of interest. For comparison, the same search task was used with the optimal gains proposed in Navalpakkam and Itti (2007).

Fig. 9 shows some of the training images used for the houses while Fig. 11 shows a typical satellite image upon which our model was used to find houses. To evaluate how well the Gaussian distributions fit the underlying probability distributions, the feature values were fit using a smoothing normal kernel function with a sliding window. The results shown in Fig. 10 indicate that the distributions do in fact resemble a Gaussian distribution. However, note that in some cases the distributions are highly peaked, which suggests that a super-Gaussian model may provide a slightly better fit.

As can be seen in Fig. 11, not all attended locations fell within houses, but the majority of locations did. On an average it took 1.52 searched locations with a standard deviation of 2.95 to find a house. The optimal gains method found a house within 1.95 searched location on average with a standard deviation of 1.51. Fig. 12 shows the percentage of the image that needed to be searched in order for find the houses in all 20 images. These results show that on average after searching about 25% of the image, all houses were found.

Fig. 12 also shows that the optimal gains method performed slightly better when finding the first few houses, but took much longer than our method to discover the more difficult targets. This slight improvement in initial performance is likely due to the fact that the optimal gains model considers both the target's and distractors' features in order to compute the best gain values. On the other hand, the SalBayes method only uses knowledge from the object to find the object. After finding a few houses, the performance of the optimal gains model drops considerably. This is mainly due to the max normalization method (see Section 2.1 and Navalpakkam & Itti, 2007 for details), which allows features which 'pop out' from the scene, yet are unrelated to the targets, to compete with those targets whose features are less visually unique.

4. Discussion

In this paper, we have developed a new unified model of attentional guidance and recognition which exploits the duality

Fig. 9. Training images used to find houses in the satellite images. The system was trained with 38 instances of houses obtained from 10 satellite images 786×786 pixels in size.

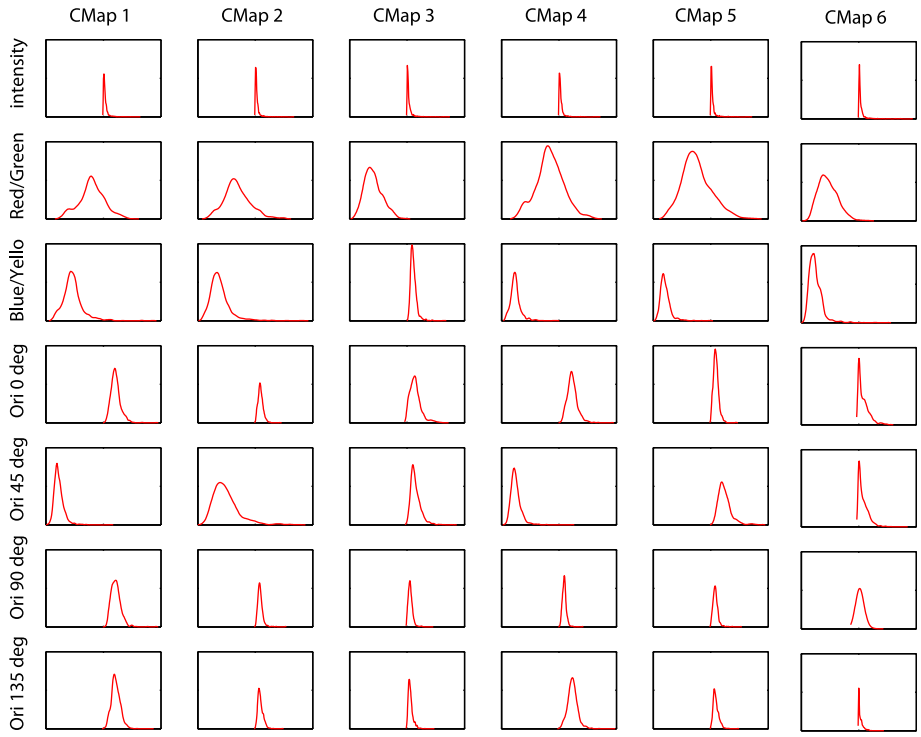

Fig. 10. Probability distribution of the houses for the various feature maps using a smoothing normal kernel function with a sliding window. The features are broken down in a grid where the rows indicate the feature type (intensity, color opponency (red–green, blue–yellow) and four orientations 0°, 45°, 90°, 135°), while the columns indicate the scale (1 being the coarsest and 6 being the finest).

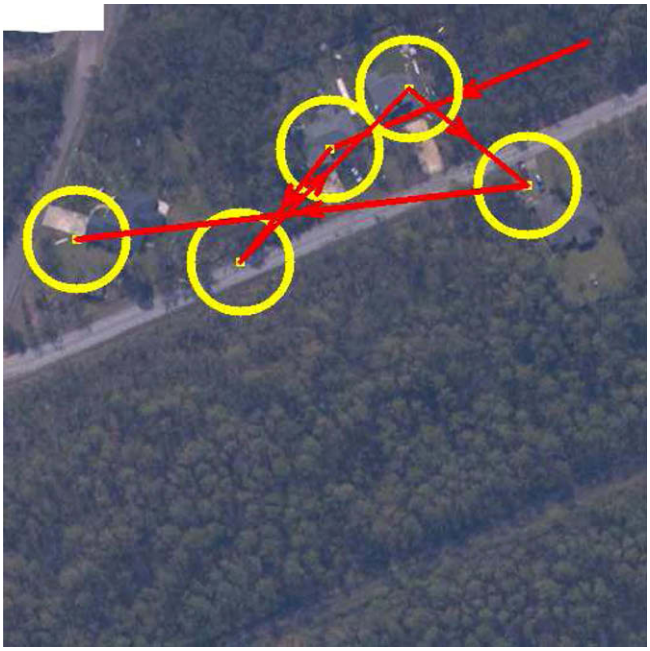

Fig. 11. Typical results for finding houses. The small yellow square indicates the fixation point, while the yellow circle indicates the inhibition of return size. The arrow shows the order in which the fixation points where chosen (which corresponded to saliency values). As can be seen, not all attended locations fell within houses, but the majority of locations did. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

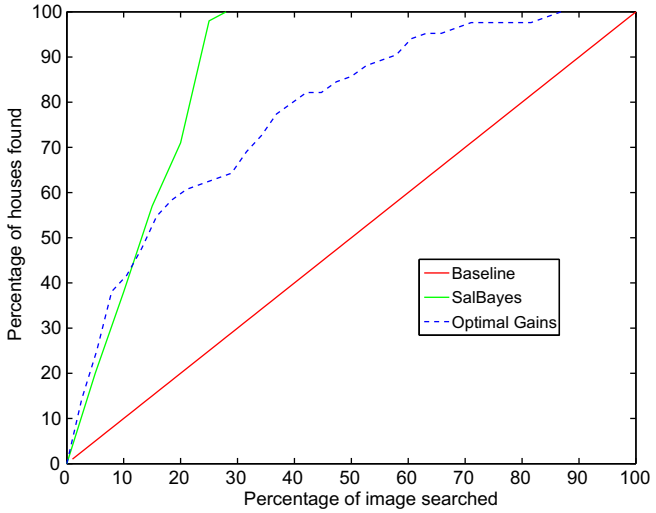

Fig. 12. Percentage of houses found vs. percentage of image searched for the 20 satellite images. Red line indicates the baseline performance if we tried to find houses at random. The green line indicates the performance achieved by SalBayes, while the dashed-blue line indicates Optimal Gains (see Navalpakkam and Itti, 2007). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

between these two tasks. On the one hand, when the model is provided with a description of an object, it will output a probability map describing the likelihood that the object can be found at each location in an input image. On the other hand, when provided with only a location in an input image, the model will provide a list of probabilities denoting the likelihood that each of it's known objects is located at the given location. As shown in the results, the model performs informed search better than previous related efforts when given difficult targets, and has shown recognition performance that is on par with current state-of-the-art methods while providing very significant speed gains.

To our knowledge, no one has extensively tested their models across the three popular datasets used here all together. The work by Murthy (2007) comes close, but they only used a subset of the ALOI dataset for recognition. As can be seen, other models have been able to achieve superior performance on specific datasets. However, it is important to note that all of the successful methods use non-parametric methods for classification which causes their computation time to grow linearly in time with the number of training views. Since the goal of the proposed model is to provide a fast first layer recognition stage, any algorithm containing complex, non-parametric classifications will not be able to efficiently support a large object database. We propose that our model could easily lend it's vast speed improvements by operating as a fast front-end to such complex algorithms, and leave the analysis of such a hybrid system to future work.

Although the model was tested using an extensive dataset of objects and scenes, additional tests using objects in natural scenes could prove useful as well. However, there have not been any datasets created thus far which contain many objects (in the order of 1000) under systematic variations embedded in natural scenes.[1] Image databases such as the LabelMe (Russell, Torralba, Murphy, & Freeman, 2005) and Caltech 101/102 (Fei-Fei, Fergus, & Perona, 2006), do not provide a systematic object search, but are more concerned with general object search. For example, finding any chair could be viewed as search, but requires much more semantic knowledge for the search (there are many types of chair that could exist). As a result, a broad semantic meaning can cause great variations and ambiguity in the search. Additionally, most of the objects that people have labeled in the LabelMe dataset are salient to begin with and would not greatly benefit from a biased saliency map (Elazary & Itti, 2008). We believe that one of the strong points of our experimental validation in this paper is that it is very systematic, which will be more difficult to achieve with these type of labeled natural scenes.

Looking at the performance in the ALOI dataset against various transformations, we see that the model does not perform as well as HMAX under the illumination color condition. This is mostly due to the fact that the model considers color information to perform classification. Therefore, as the color of the object changes (due to the color of the light) the model encounters more ambiguity. However, such changes in color illumination do not often occur in the real world, and so we claim that robustness to 3D transformation and illumination direction are more desirable features in a first level recognition system.

Looking at the timing aspects of the methods tested, it can be seen that the proposed method, SalBayes, outperforms both SIFT and HMAX by many folds. Furthermore, the time requirement for both HMAX and SalBayes does not change significantly with training datasets (both decrease as the amount of remaining testing data decreases). This results from the underlying classifier that is used to classify the features. Both SalBayes and Hmax use a parametric density function to estimate the probability of the features belonging to a particular class. However, SIFT uses a non-parametric estimation (k-NN) which results in an increase in the time required to classify a given feature with the increase in training data.

While examining the performance of the proposed model, it was found that additional training examples did not always improve performance. This can be attributed to ambiguities developed by modeling each feature distribution as a unimodal Gaussian. When too many training instances are used, the actual distribution of a feature's density function can become multi-modal, which can then be poorly approximated by the model. Future work is planned to evaluate more advanced PDF representations, such as mixtures of Gaussians or particle filters to try to accommodate for such situations. However, despite these limitations, the proposed model has shown that from a very small dimensional feature vector (42 dimensions) at a single location on an object (the most salient location), the model was successfully able to distinguish among many objects.

One improvement to the model could be made by the choice of probability distribution. For example, after examining the features of particular objects, it was found that often the feature distribution could not be simply modeled using a single Gaussian model. That is, some of the variations of particular features could not be explained with a normal distribution. In particular the color (under the color illumination changes) and the orientations (under the rotation variations). As a result, estimating this as a normal distribution would cause errors in biasing and classifying the features. One explanation for the shape of these distributions can be due to the various ranges of values for different objects of the same class. For example, an object could contain strong red features and weak red features depending on the illumination color.

It was also found that the distributions in the ALOI dataset often exhibited two modes (which were primarily due to the changes in orientations and changes in illuminations). If the various variations of the objects can be modeled, then a single Gaussian can be used to describe a particular part of an object, and the mixture can be used to describe all the parts. Therefore, using a mixture of Gaussian model can provide a better model of the probability distribution. Training the mixture of Gaussian can be achieved by using an expectation–maximization (EM) algorithm. The drawbacks of this algorithm, however, are that it is an iterative method and requires that all training exemplars be available in each iteration. It would be worth investigating how the mixture of Gaussian model can be learned on-line as new inputs come in. One suggested way would be to cluster the data, extract the means, and then learn a single Gaussian on the cluster. The multiple clusters would then yield the mixture model.

Fig. 10 shows that some of the distributions in the satellite images house search could have been modeled using a super-Gaussian, to account for the sharp peak in the distribution. For example, the Laplace or logistic distribution could have been used in some of the distributions to model this peak. The results of which can improve performance by not only increasing the probability around the mean but accounting for more variations by having a fatter tail. However, future research will need to determine when and how to switch distribution models and how will this effect performance for both searching and recognition.

Examining the satellite images search results (Fig. 12), we see that the performance of the Optimal Gains proposed in Navalpakkam and Itti (2007) performs the same as the proposed model for the first few houses, but then loses performance when attempting to find more houses. The reason is that the Optimal Gains follows a similar structure proposed by Treisman's Feature Integration Theory (Treisman & Gelade, 1980; Treisman & Sato, 1990) and Wolfe's Guided Search (Wolfe, 1994) in which whole spatial maps of feature detectors are biased towards the target. Considering the neural hardware available in the brain (each neuron can perform computations independent of each other), it could be conceived that each neuron can be biased separately, which is the approach we have chosen to take in this paper. Additionally, we bias the feature maps with more of a probabilistic approach (applying a PDF for each neuron) as opposed to a simple gain change. This would enable the system to bias for weak features among strong ones as discussed in the introduction (since applying a gain would boost features and not suppress them). From a biological aspect, this can be seen as shaping the profiles of detectors that are more likely to respond to the target by shaping their tuning curve toward the tar-

[1] We are currently in the process of building an extensive dataset where the same objects are photographed in different complex natural backgrounds under different light conditions and poses.

get individually, using prior knowledge about the object. This results in great granularity in the discrimination ability of the search without the added overhead and limitations of multiple sub-bands.

Additionally, it is important to note that the optimal gains system as well as Feature Integration Theory and Guided Search is trained for search specifically, and does not uses this information for classifying the object. In this paper, we concentrated on the synergy between learning the parameters for the classification, and then using them for search. However, a hybrid system could be used so that the object can be searched more efficiently in the presence of known distractors. In particular, using some of the knowledge of the distractors could help achieve grater performance under certain conditions.

Lastly, the model proposed in this paper works in situations in which the object can be described using few simple features. For example, a house or a road can be described using simple features. However, more complex objects or scenes would need multiple features spanning greater spatial distance (more than the fovea size) to be described. For example, an urban area does not only contain a house but also contains multiple houses and roads (as seen from above). As a result, it would be advantageous if more knowledge can be added to the biasing, as proposed by Navalpakkam and Itti (2005). This knowledge would describe the parts of the object, and its relation in the scene. For example, if the system is looking for a refrigerator, then it knows that refrigerators are composed of doors. In addition, if the system is looking at a kitchen scene, then it can first check the likely locations of fridges within the scene first. Therefore, the knowledge of scenes can be used to efficiently speed up the search in more complex scenes. This knowledge can also boost the recognition rate by setting up the appropriate prior for the scene. For example, if we know the probability of a fridge appearing in a given scenes, ambiguities in appearances with another objects (say a door) can be resolved using the prior information about the scene. This knowledge can be provided from gist models, such as one proposed by Torralba, Oliva, Castelhano, and Henderson (2006). In addition, the knowledge base can be used to narrow down the search for features. For example, if a few houses are already encountered, then the system should check for the presence of trees. Therefore, the next fixation should bias for trees. As a result, the system knows that this could not be an ocean (because of the structure in the knowledge base), so it should not bias for boats. For a previous implementation of such a system (see Navalpakkam & Itti, 2005).

Acknowledgments

This work was supported by the Army Research Office (ARO), the Human Frontier Science Program (HFSP), the National Science Foundation (NSF), the Defense Advanced Research Projects Agency (DARPA), and the National Geospatial-Intelligence Agency (NGA). The views, opinions, and findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency, National Geospatial-Intelligence Agency, or the U.S. Department of Defense.

Appendix A

A.1. HMAX

This visual structure was first proposed by Riesenhuber and Poggio (1999) and later improved by Serre et al. (2005). It was dubbed HMAX ("Hierarchical Model and X") and has drawn its inspiration from biological vision. The main contribution of the structure is its ability to achieve invariance at the local level by pooling local features using a max operator in both scale and position. The whole structure is built from two layers, where the first layer extracts Gabor features and pools them together. The pooling first takes the max over the position by sub-sampling the space into a grid size N-band and then taking the max between scales. The second layer extracts codewords at random from the first layer and stores them in a database. The response of the layer is then computed by a distance measure between the memorized patches and the current stimulus using Radial Basis function (RBF). Lastly, an SVM is used to classify objects based on the features from the second layer.

A.2. SIFT

This algorithm has been proposed by Lowe (2004) and is known as SIFT, which stands for Scale-Invariant Feature Transform. The algorithm first extracts keypoints by using local scale-space maxima and minima of various Difference of Gaussian (DoG) operations applied to the input image. This results in keypoints from varius locations and scales with heigh texture energy. From these keypoints, a descriptor vector invariant to scale, translation, slight 3D rotations and intensity is created. This is achieved with a 128 dimensional vector indicating the gradient locations and orientations using a histogram. The space is quantizes into a 4×4 grid while the orientations are quantized into eight orientations. These descriptor vectors are store in a database for classification.

During the classification stage, the same processes described above is used to extract various descriptor vectors from a new image, while a Nearest Neighbor algorithm is used to find matches in the database. Additional, at least three close matching keypoints are required to match with an additional affine constraint (checked with an Hough transform) in order for the object to be recognized.

A.3. SVM

Support vector machines (SVM) are a method of supervised classification and regression first proposed by Vladimir Vapnik in 1963 for linear separation. The hypothesis space of an SVM is a set of hyperplanes that attempts to achieve the largest distance to any sample in the training dataset for any class, which is known as the functional margin. To handle non-linear classification, SVMs employ a kernel trick proposed by Boser et al. (1992), which first maps the data into a liner space using a kernel of some kind, and then performs the linear separation. Common kernels include Polynomial, Radial Basis Function and Gaussian functions.

References

Barfoot, T. (2005). Online visual motion estimation using fastslam with sift features. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS), Edmonton, Alberta, 2–6 August 2005* (pp. 3076–3082).

Bonaiuto, J., Itti, L. (2005). Combining attention and recognition for rapid scene analysis. In *Proceedings of IEEE-CVPR workshop on attention and performance in computer vision (WAPCV'05), San Diego, California, June 2005* (pp. 1–6).

Boser, B. E., Guyon, I. M., Vapnik., V. N. (1992). A training algorithm for optimal margin classifiers.

Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal (Q2)*, 15. <http://citeseer.ist.psu.edu/bradski98computer.html>.

Burianek, J., Ahmadyfard, A., Kittle, J. (2001). Soil-47 the surrey object image library. <http://www.ee.surrey.ac.uk/Research/VSSP/demos/colour/soil47/>.

Chang, C.-C., Lin, C.-J. (2001). LIBSVM: A library for support vector machines. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.

Comaniciu, D., Meer, P. (1997). Robust analysis of feature spaces: Color image segmentation. <http://citeseer.ist.psu.edu/comaniciu97robust.html>.

Desimone, R., & Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annual Review of Neuroscience, 18*, 193–222.

Elazary, L., & Itti, L. (2008). Interesting objects are visually salient. *Journal of Vision, 8*(3:3), 1–15.

Elinas, P., & Little, J. J. (2005). sMCL: Monte-Carlo localization for mobile robots with stereo vision. In S. Thrun, G. S. Sukhatme, & S. Schaal (Eds.), *Robotics: Science and*

systems I, June 8–11, 2005 (pp. 373–380). Massachusetts Institute of Technology, Cambridge, Massachusetts: The MIT Press. <http://www.roboticsproceedings.org/rss01/p49.html>.
Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(4), 594–611.
Geurts, R. P., Piater, J., Wehenkel, L. (2004). A generic approach for image classification based on decision tree ensembles and local sub-windows. In: *Proceedings of 6th Asian conference on computer vision* (pp. 860–865).
Geusebroek, J. M., Burghouts, G. J., & Smeulders, A. W. M. (2005). The Amsterdam library of object images. *International Journal of Computer Vision, 61*(1), 103–112.
Gonzalez, R. C., & Wintz, P. (1987). *Digital image processing* (2nd ed.). Addison-Wesley.
Hayhoe, M., & Ballard, D. (2005). Eye movements in natural behavior. *Trends in Cognitive Sciences, 9*(4), 188–194. <http://dx.doi.org/10.1016/j.tics.2005.02.009>.
Horn, B. K. P. (1986). *Robot vision*. Cambridge, MA: MIT Press.
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.
Itti, L., & Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews Neuroscience, 2*(3), 194–203.
Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(11), 1254–1259.
Kittler, J., & Ahmadyfard, A. (2001). On matching algorithms for the recognition of objects in cluttered background. In *Proceedings of the 4th international workshop on visual form (IWVF-4)* (pp. 51–66). London, UK: Springer-Verlag.
Krummenacher, J., Muller, H., Reimann, B., & Heller, D. (2001). Visual singleton detection ('pop-out') is mediated by dimension-based attention. In *Proceedings 17th annual meeting of the international society for psychophysics* (pp. 479–486). Pabst Science Publishers.
Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV* (pp. 1150–1157). <http://dx.doi.org/10.1109/ICCV.1999.790410>.
Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110.
MacLean, W. J., & Tsotsos, J. K. (2008). Fast pattern recognition using normalized grey-scale correlation in a pyramid image representation. *Machine Vision Applications, 19*(3), 163–179.
Maree, R., Geurts, P., Piater, J., & Wehenkel, L. (2005). Random subwindows for robust image classification. *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 34–40). Washington, DC, USA: IEEE Computer Society. <http://dx.doi.org/10.1109/CVPR.2005.287>.
Mikolajczyk, K., Leibe, B., & Schiele, B. (2005). Local features for object class recognition. *ICCV, 2*, 1792–1799.
Moran, J., & Desimone, R. (1985). Selective attention gates visual processing in the extrastriate cortex. *Science, 229*(4715), 782–784.
Motter, B. (1994). Neural correlates of attentive selection for color or luminance in extrastriate area v4. *Journal of Neuroscience, 14*, 2178–2189.
Murthy, C. A. (2007). Distinct multicolored region descriptors for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(7), 1291–1296 (member – Sarif Kumar Naik).
Navalpakkam, V., Itti, L. (2006a). An integrated model of top-down and bottom-up attention for optimal object detection. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), New York, NY, June 2006* (pp. 2049–2056).
Navalpakkam, V., & Itti, L. (2005). Modeling the influence of task on attention. *Vision Research, 45*(2), 205–231.
Navalpakkam, V., & Itti, L. (2006b). Top-down attention selection is fine-grained. *Journal of Vision, 6*(11), 1180–1193.
Navalpakkam, V., & Itti, L. (2007). Search goal tunes visual features optimally. *Neuron, 53*(4), 605–617 (also see commentary/preview entitled "Paying Attention to Neurons with Discriminating Taste" by A. Pouget, D. Bavelier, Neuron 2007;53(4):473–475).
Nene, S. A., Nayar, S. K., Murase, H. (1996). Columbia object image library (coil-100).
Obdrzalek, S., Matas, J. (2002). Object recognition using local affine frames on distinguished regions. In Rosin, L. Paul, David Marshall (Eds.), *Proceedings of the British machine vision conference* (Vol. 1, pp. 113–122).
Obdrzalek, S., Matas, J. (2005). Sub-linear indexing for large scale object recognition. In *BMVC05* (pp. 113–122).
Pratt, W. (1991). *Digital image processing* (2nd ed.). New York: Wiley.
Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience, 2*, 1019–1025.
Rish, I. (2001). An empirical study of the naive bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*.
Roth, D., Yang, M., & Ahuja, N. (2002). Learning to recognize three-dimensional objects. *Neural Computation, 14*(5), 1071–1103.
Russell, B., Torralba, A., Murphy, K., Freeman, W. (2005). Labelme: A database and web-based tool for image annotation. *MIT AI Lab Memo AIM-2005-025, September 2005*.
Scott, D. W. (1992). *Multivariate density estimation*. Wiley.
Se, S., Lowe, D., Little, J. (2002). Global localization using distinctive visual features (October 2007).
Serre, T., Wolf, L., Poggio, T. (2005). Object recognition with features inspired by visual cortex. *Computer vision and pattern recognition (CVPR 2005), San Diego, USA, June 2005*.
Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., & Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(3), 411–426.
Shokoufandeh, A., Marsic, I., Dickinson, S. (1998). View-based object recognition using saliency maps. <citeseer.ist.psu.edu/48145.html>.
Sim, R., Elinas, P., Griffin, M. (2005). Vision-based slam using the raoblackwellised particle filter. In *Proceedings of IJCAI workshop on reasoning with uncertainty in robotics*.
Snavely, N., Seitz, S., Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3d. In *ACM transactions on graphics* (pp. 835–846). New York, NY, USA: ACM Press.
Tagare, H. D., Toyama, K., & Wang, J. G. (2001). A maximum-likelihood strategy for directing attention during visual search. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(5), 490–500.
Theeuwes, J. (1994). Endogenous and exogenous control of visual selection. *Perception, 23*, 429–440.
Theeuwes, J. (1995). Abrupt luminance change pops out; abrupt color change does not. *Perception & Psychophysics, 57*(5), 637–644.
Torralba, A., Oliva, A., Castelhano, M., & Henderson, J. M. (2006). Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review, 113*, 766–786.
Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology, 12*(1), 97–136.
Treisman, A., & Sato, S. (1990). Conjunction search revisited. *Journal of Experimental Psychology – Human Perception and Performance, 16*(3), 459–478.
Treue, S., & Trujillo, J. M. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature, 399*, 575–579.
Tsotsos, J. K. (1991). Computational resources do constrain behavior. *Behavioral and Brain Sciences, 14*(3), 506.
Vasconcelos, M., & Vasconcelos, N. (2009). Natural image statistics and low-complexity feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(2), 228–244.
Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review, 1*, 202–238.
Wolfe, J. M. (1998). Visual memory: What do you know about what you saw? *Current Biology, 8*(9), R303–R304.
Wolfe, J. M., & Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience, 5*(6), 495–501.