Optimal Scanning for Faster Object Detection

Nicholas J. Butko
UC San Diego, Dept. of Cognitive Science
La Jolla, CA 92093-0515
nbutko@cogsci.ucsd.edu

Javier R. Movellan
Institute for Neural Computation
La Jolla, CA 92093-0523
movellan@mplab.ucsd.edu

Abstract

Recent years have seen the development of fast and accurate algorithms for detecting objects in images. However, as the size of the scene grows, so do the running-times of these algorithms. If a 128×102 *pixel image requires 20ms to process, searching for objects in a* 1280×1024 *image will take 2s. This is unsuitable under real-time operating constraints: by the time a frame has been processed, the object may have moved. An analogous problem occurs when controlling robot camera that need to scan scenes in search of target objects. In this paper, we consider a method for improving the run-time of general-purpose object-detection algorithms. Our method is based on a model of visual search in humans, which schedules eye fixations to maximize the long-term information accrued about the location of the target of interest. The approach can be used to drive robot cameras that physically scan scenes or to improve the scanning speed for very large high resolution images. We consider the latter application in this work by simulating a "digital fovea" and sequentially placing it in various regions of an image in a way that maximizes the expected information gain. We evaluate the approach using the OpenCV version of the Viola-Jones face detector. After accounting for all computational overhead introduced by the fixation controller, the approach doubles the speed of the standard Viola-Jones detector at little cost in accuracy.*

1. Introduction

Detecting objects quickly and at low computational cost is important for a wide variety of domains, such as security applications, traffic analysis, clinical diagnosis, satellite image processing, and robotics. While progress in recent years has been dramatic, there are still two challenging cases: (1) Physical scanning of scenes using active cameras, and (2) Digital scanning of very large images. For scanning scenes using active visual sensors, biology has chosen a solution based on the use of foveal sensors whose resolution diminishes as a function of eccentricity. Scanning very large images can be seen as a special case of scanning world scenes. Thus it is reasonable to expect that the approaches that biology has found useful for scanning the world may also be useful for scanning high resolution images. In this paper we explore this idea by digitally simulating in software a "foveal camera". The sequential placement of the digital fovea is then controlled using a policy designed to maximize the information gathered about the location of the target of interest. The proposed approach is "plug-and-play": it can be applied to standard object detectors in a modular manner. In this, our first implementation, we double the computational efficiency of current object detectors. I.e., the computational overhead required to implement the digital fovea and control policy is more than compensated by the improvements in scanning efficiency. The source code needed to reproduce the results in this paper is provided online as part of Nick's Machine Perception Toolbox [3].

1.1. Digital Fovea

Key to the proposed approach is the idea of scanning images using a simulated fovea. Given a fixation point of the virtual camera, the simulated fovea yields a collection of Image Patches (IP) of different sizes, all of them centered on the fixation point (see Figure 1). Each of the IPs is then shrunk to a common reference size that is much smaller than the original image. These different patches will lose information about the image in different ways: IPs larger than the reference size may cover most of the image, but they will lose resolution when scaled down to the smaller reference size. IPs smaller than the reference size will maintain resolution but only around a small region of the image. Due to the fact that all the patches are centered at a fixation point, the consequence is that resolution is preserved around the fixation point, but falls off in the periphery, thus the name "digital fovea."

Figure 1 shows an example of the digital fovea at work. In this case we used 4 IPs per fixation, thus operating at 4 scales. To search for the target object at that fixation point, we can apply any off-the-shelf object detection algorithm

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

to each of these IPs. The object detector will search each of the IPs exhaustively for the target object. E.g. a Viola Jones style detector will search each downsampled IP at all locations and scales. As long as the scaled size of the Image Patches is small, this exhaustive search will be quick.

For example: If any IP is scaled to 10% of the height and width of the image, its area is 1% of the original image. Since all 4 IPs are shrunk to the same small size, an object detector with linear complexity will search all 4 IPs in 4% of the time it would take to search the whole image. If the search target's location can be inferred after scanning IPs at fewer than 25 successive fixations, this foveated approach will be faster than exhaustively applying object detection to a high resolution image.

Two particular challenges are: (1) sequentially picking the fixation locations; (2) integrating the information acquired during each successive fixation. The problem of optimal information gathering and integration is a standard (but basically unsolved) problem in stochastic optimal control. The nature of this problem is similar to that faced by humans when moving their eyes, so we turn to the literature on human eye-movements to guide our approach.

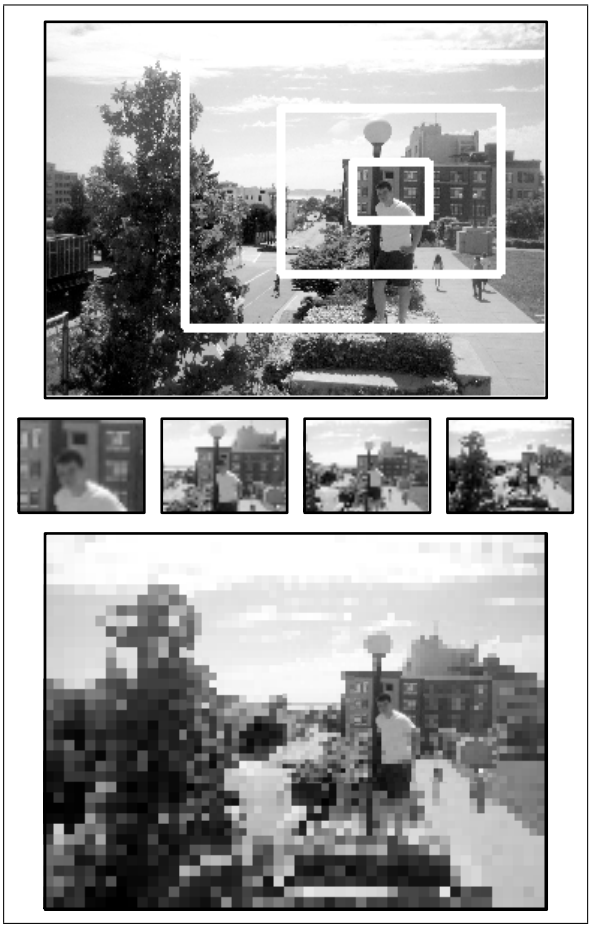

Figure 1. A digital fovea: Several concentric Image Patches (IPs) (*Top*) are arranged around a point of fixation. The image portions contained within each rectangle are reduced to a common size (*Middle*). In a reconstruction from the downsampled images, detail is preserved around the fixation point, but decreases with eccentricity (*Bottom*).

1.2. Related Work

Our work relates to the growing literature on computational approaches to eye movements and visual saliency. Models of visual saliency [13, 8, 18] have been shown to provide a useful way to improve the search efficiency of specific object detectors, i.e., most regions without objects tend to have low visual saliency [5]. Unfortunately visual saliency filters are computationally expensive [17] and need to be applied to entire images, making them less attractive for scanning very high resolution images.

Our work also relates to recent work on optimal image search, like the Efficient Subwindow Search [10]. Our approach is data driven and detector independent, where the ESS approach is more analytic. Our approach requires a dataset of labeled images to build a statistical model of the performance of a given object detector. The ESS approach requires a function $\hat{f}$ that must be constructed analytically for each specific object detector for the guarantees of the algorithm to hold, but only some object detectors are amenable to such a construction. The efficiency of the algorithm depends on the tightness of the upper bound that $\hat{f}$ computes and the computational overhead of evaulating $\hat{f}$.

2. I-POMDP: A Model of Eye-Movement

Najemnik & Geisler developed an information maximization (Infomax) model of eye-movements and applied it to explain visual search of simple objects in pink noise image backgrounds [12]. The model uses a greedy search approach: saccades are planned one at a time with the next saccade made to the location in the image plane that is expected to yield the highest chance of correctly guessing the target location. The Najemnik & Geisler model successfully captured some aspects of human saccades but it has two important limitations: (1) Its fixation policy is greedy, i.e., it maximizes the instantaneous information gain rather than the long term gathering of information. (2) It is applicable only to artificially constructed images.

Butko & Movellan [4] proposed the I-POMDP framework for modeling visual search. The framework extends the Najemnik & Geisler model by applying long-term POMPDP planning methods. They showed that long-term information maximization reduces search time. Moreover the optimal search strategy varies in principled ways with the characteristics of the optical device (e.g. eye vs. camera) that is used for searching [4]. While this addressed the

first limitation of the Najemik & Geisler model, the second limitation remained unaddressed, i.e, the model was only suitable for a limited class of psychophysical stimuli, namely images that can be described as containing point objects in a field of Gaussian noise. In this document, we present a first attempt to extend the I-POMDP model to be useful for computer vision applications.

I-POMDP frames visual search as a *Partially Observable Markov Decision Process* (POMDP) [9]. A POMDP can be described as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, R, P_T, P_O \rangle$. The sets $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{O}$ describe the possible States, Actions, and Observations of the POMDP. R is a reward function that describes the goal. P_T and P_O are probability distributions that describe the State-Transition dynamics, and the State-Observation probabilities respectively. The State is not directly observable, but can be inferred from sequential Actions and Observations.

In the I-POMDP framework a visual target is located at one of N discrete locations, arranged on a grid. The State $S \in \mathcal{S} = [1, 2, ..., N]$ describes the current grid location of the target. The Action $A \in \mathcal{A} = [1, 2, ..., N]$ describes which grid location the subject is currently fixating. The observation vector $\vec{O}_t \in \mathcal{O} = \mathcal{R}^N$ consists of some noisy, real-valued information from each grid point about whether the target is present or absent there collected during the fixation at time t. An element O_t^i of the vector corresponds to grid-point i.

Each observation vector $\vec{O}$ is drawn from the conditional probability distribution $P_O(\vec{O}|S, A)$ that follows a "Signal Plus Noise" paradigm. In the original version of I-POMDP each pixel response is modeled as the combination of two processes: an i.i.d. Gaussian noise process, and, if the pixel renders the target, a signal process. The strength of the signal depends on the eccentricity of the pixels with respect to the current fixation point. The relationship between eccentricity and signal determines the Fovea-Periphery Operating Characteristic function, $F(||S, A||)$.[1] The observation generation model, depicted graphically in Figure 2, gives

$$\begin{aligned} P_O(\vec{O}_t = \vec{o}_t \,|S_t = i, A_t = k) = \\ = N(o_t^i; \mu = F(||i, k||), \sigma^2 = 1) \\ \prod_{j \neq i} N(o_t^j; \mu = 0, \sigma^2 = 1) \end{aligned} \quad (1)$$

where $N(o_t^j; \mu, \sigma^2)$ is the Gaussian likelihood of the specific value of o_t^j given the parameters μ and σ^2, and $||i, k||$ is the euclidean distance between grid points i and k.

Each fixation provides new information which is used to update the system's beliefs about the location of the target, i.e., the posterior distribution of the target given the history

[1]Najemnik & Geisler estimated this curve psychophysically in their subjects. [12]

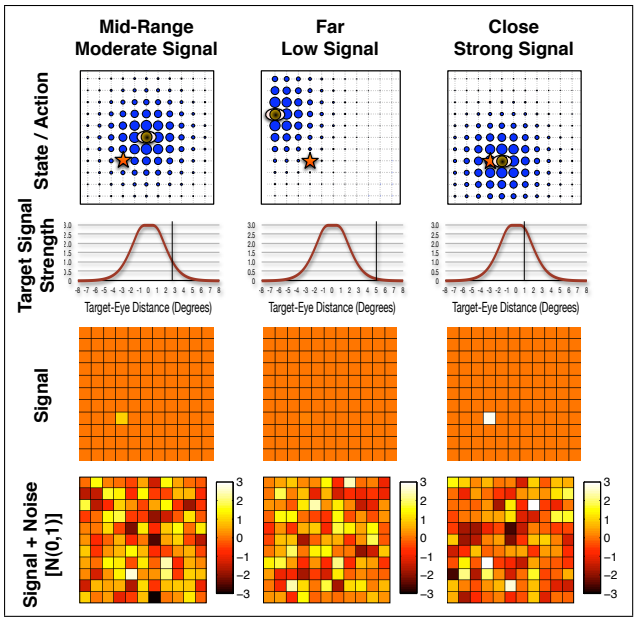

Figure 2. The I-POMDP model of Eye-Movement: A target is located at a visual location previously unknown to the subject. When the subject observes the world, unit-Gaussian sensor noise corrupts the observation. When the subject is looking close to the target, the target gives off a strong signal, while when the subject looks far away, the signal is weak. By making several fixations and integrating observations across fixations, the subject eventually becomes confident in the location of the visual target.

of observations. This is done using standard Bayesian inference. The subject's belief B_t^i about how likely it is that the search target is located at grid-position i can be written as follows

$$B_t^i \propto p(\vec{O}_t|S_t = i, A_t = k) B_{t-1}^i \quad (2)$$

$$= \left[\prod_{j=1}^{N} p(o_t^j|S_t = i, A_t = k) \right] B_{t-1}^i \quad (3)$$

$$\propto \frac{p(o_t^i|S_t = i, A_t = k)}{p(o_t^i|S_t \neq i, A_t = k)} B_{t-1}^i \quad (4)$$

where (3) follows from (2) by the independence in sensor noise, and (4) follows by noticing that the probability that the entire observation vector was generated only by the noise process is a constant, i.e. $\prod_j p(o_t^j|S_t \neq j, A_t = k) = C_k$. The goal in I-POMDP is to develop a policy that maps the current belief state (the posterior distribution of the target location) into actions (next fixation). This policy is designed to maximize the long-term gathering of information about the target location. This is equivalent to minimizing the entropy of the belief distribution $\vec{B}_t$ [11]. Thus the reward function at time t is the negative entropy of the poste-

rior distribution at that time:

$$R(\vec{B}_t) \quad = \quad \sum_{i=1}^{N} B_t^i \log B_t^i \tag{5}$$

The measure of how well a given policy is gathering information is the reward accrued across a potentially infinite number of fixations: $\sum_{t=0}^{\infty} \gamma^t R(\vec{B}_t)$, where $0 < \gamma < 1$ is the discount factor. While this appears to be a very complex control problem, it has strong constraints, e.g., shift invariance, that make possible the efficient use of stochastic optimization methods, like Policy Gradient [1].

As presented here the I-POMDP model assumes that there is exactly one target in the image plane. It is straightforward to extend the I-POMDP model to the case where there is *at most* one search target by adding a special state, $S_t = 0$ indicating that no target is present. The belief update for this state is $B_t^0 \propto 1$ given the update rule in (4). Extending the algorithm to multiple targets in a principled manner is tricky. In practice if there are multiple targets, either the algorithm will only discover one of them, or it will assign approximately equal probability to the two target locations.

2.1. The Multinomial I-POMDP Model

While I-POMDP provided a principle approach to image search, it was limited to a very restricted class of images thus rendering it not useful for realistic computer vision applications. Here we present a variant of the original I-POMDP framework, named Multinomial I-POMDP (MI-POMDP), that can be easily applied to off-the-shelf object detectors, like the Viola-Jones face detector [16, 15].

State: In I-POMDP, the state $S_t = i \in \mathcal{S} = [1, 2, ..., N]$ indicates that the search target is located at the grid location i. This abstract state representation needs to be made concrete for object detection in images. Concretely, we cover the image with a discrete grid, and assume that the location of the object's center is inside one of those grid locations. A natural tradeoff arises in choosing how fine to make the grid: A finer grid groups fewer pixels into each grid cell, improving the ability to localize the object in the image; but this increases the number of hypotheses that must be entertained and locations that can be searched. For this paper we chose to tile the image with a 21×21 grid, meaning the search target could be located at any of 441 locations. This discretization can be seen in Figure 3. Depending on the size of the image, more or fewer pixels may be grouped into each grid cell.

Action: In I-POMDP, action $A_t = i \in \mathcal{A} = [1, 2, ..., N]$ indicates the current center of fixation; the effect of fixation was encoded in the $F(||i, k||)$, which describes how the search target signal dropped as a function of distance from fixation. For digital foveas, a similar effect is achieved by effectively decreasing the resolution with increasing distance from fixation. In practice this is achieved by the mechanism of a pyramid of IPs [7].

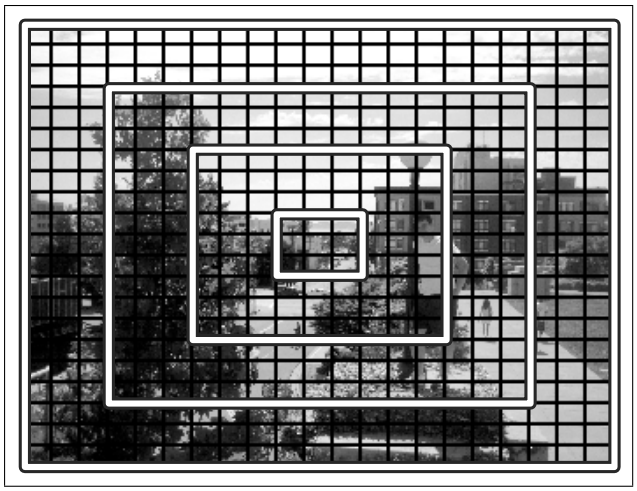

Figure 3. A 21×21 grid was laid over each image, forming the basis of the hypotheses that are entertained about the possible location of a face in the image. A pyramid of concentric Image Patches (IPs) surround the current point of fixation, which in this example is the central grid-cell.

Any grid-point can be the center of fixation, marking the center of the IP pyramid. IPs of several scales are placed concentrically around the fixation point. We used a pyramid of 4 IPs with diameters of 3, 9, 15, and 21 grid-cells. An example of fixating the center of the image is shown in Figure 3. If an IP could not be placed concentrically around the fixation-point without being partially off the image, it was stopped at the image border and so was effectively off-center from the fixation. This way, each IP was completely filled with part of the image. An example of an off-center IP is in Figure 1 where the third-smallest scale is stopped by the right edge of the image. Its center is to the left of the point of fixation.

Observation & Observation Model: A probabilistic model of Observations and how they are generated is important for deducing the target location with Bayesian inference. A major challenge is to turn the output of the object detector into a suitable observation vector. We treat object detectors as black-box algorithms that take an image as input, and output a list of pixels that are likely to be the centers of the search target. These detectors often fire in clusters around the object (hits), but also have false alarms, misses, and correct rejections (Figure 4). In MI-POMDP, the observation is the total number of objects returned by the object detector in each grid cell (up to some maximum count value, C_{max}), after searching all IPs. The observation vector generated is $\vec{O}_t \in \{0, 1, ..., C_{max}\}^N$.

Because information is lost in the digital fovea, there is uncertainty about whether the object detector will find the

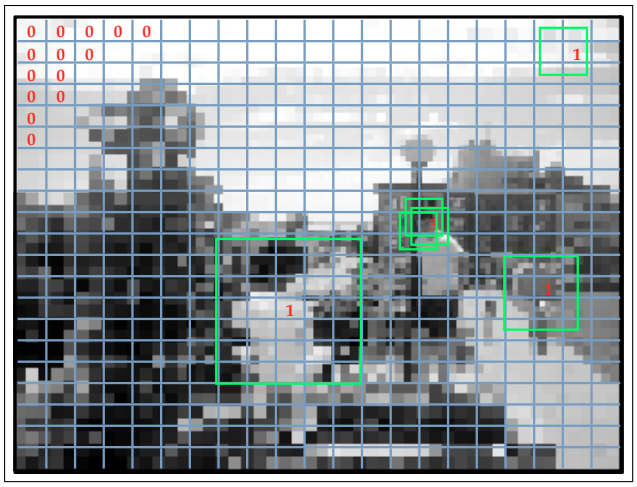

Figure 4. An object detector returns candidate locations of the search target. In each grid cell, we count the candidates up to some maximum (above, empty cells have an observation of "0"). We model the counts as being generated by independent draws from many multinomial distributions, with parameters that vary with the distance to the point of fixation, and also whether the search target is actually centered at that grid cell.

object (false negative); given that an object detector finds an object, it is uncertain whether this is actually the object (false positive). We represent this uncertainty by modeling the generation of each grid cell's contribution to the observation vector as an independent draw from a different Multinomial distribution conditioned on: 1) the presence or absence of an object in that grid cell; 2) The distance (x-distance and y-distance) to the center of fixation from that grid cell. Practically, this means for an $L \times M$ grid of target locations, each observation is drawn from one of $2LM$ multinomial distributions with different parameters for each combination of x-distance $\in [0, 1, ..., M-1]$, y-distance $\in [0, 1, ..., L-1]$, and object presence / absence.

System Dynamics: In images, the target we are searching for does not move, and the POMDP belief update equation in Equation (4) can be used. In active cameras or video streams, the target might move between each fixation. In this case, the dynamics are modeled by $p(S_t = i|S_{t-1} = h)$, and the belief update becomes

$$B_t^i \propto \frac{p(o_t^i|S_t = i, A_t = j)}{p(o_t^i|S_t \neq i, A_t = j)} \sum_{h=1}^{N} p(S_t = i|S_{t-1} = h) B_{t-1}^h \quad (6)$$

For further discussion, see Section 5.

3. Implementation

The MI-POMDP model is framed in general formalisms that are agnostic to the object being searched for, or for the detector given. We tested it with the OpenCV 1.0 face detector, a Viola-Jones style face detector [15, 16]. For this paper we chose to tile all images with a 21×21 grid, meaning the face could be localized to any of 441 locations. We used IPs with diameters of 3, 9, 15, and 21 grid-cells. When the smallest IP was smaller than 60×45 pixels, it was not used. The downsampled image size was always the same number of pixels as the smallest IP used. The full source code needed to implement this model is provided online as part of Nick's Machine Perception Toolbox [3].

3.1. Image Dataset

We evaluated our algorithm using images from the GENKI2005 dataset of over 50,000 images of faces [6]. In GENKI2005, most faces were a significant fraction of the image plane, making them quite easy to search for (by searching large image scales first). To increase the difficulty, we selected a subset of 3,500 images randomly such that faces were present in equal amounts across all scales. Specifically, $\frac{1}{5}$th were $< 10\%$ of the image major axis, and $\frac{1}{5}$th each were 10-20%, 20-30%, 30-40% and 40%+ of the image major axis. The full images varied in size from 104×120 to 972×477 with an average size of 225×243. This new data set is freely available as the size-scale normalized subset (GENKI-SZSL) of the GENKI dataset [14].

3.2. Fitting the Multinomial Observation Model

The observation model presented above consists of $2LM$ multinomial distributions, each with $C_{max} + 1$ differently weighted outcomes. To fit the model, we estimated the weights for each outcome for each distribution, using $C_{max} = 9$.

We started with a $2 \times 21 \times 21 \times 10$ table T filled with ones. For each image in the dataset, we fixated the digital fovea on every grid point k, and computed C, the count of found face boxes centered in each grid cell up to $C_{max} = 9$. On each fixation, for each of the 440 locations j without a face, we computed $XDist(j,k)$ and $YDist(j,k)$ from that location to the point of fixation, and incremented the table element $T[0, XDist(j,k), YDist(j,k), C]$. For the one location i with a face, we incremented the table element $T[1, XDist(i,k), YDist(i,k), C]$.

After this procedure, the estimates

$$P(O_j = C|S \neq j, A = k) = = \frac{T[0, |XDist(j,k)|, |YDist(j,k)|, C]}{\sum_{C'=0}^{C_{max}} T[0, |XDist(j,k)|, |YDist(j,k)|, C']} \quad (7)$$

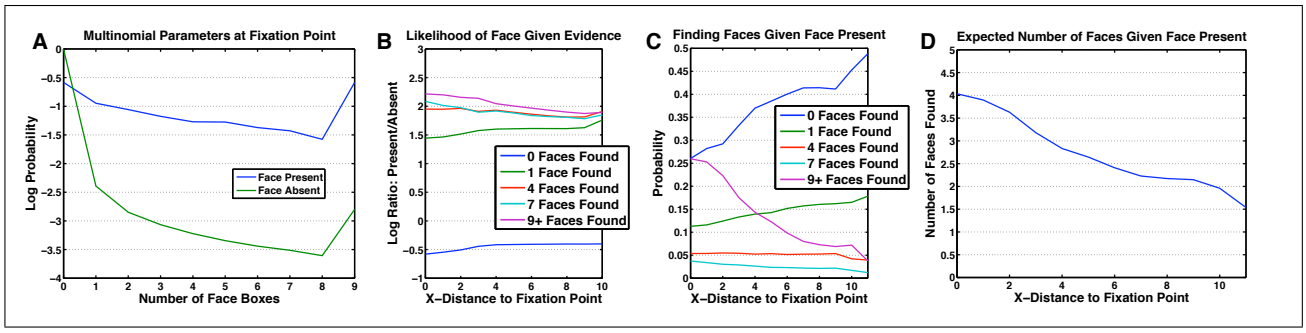

Figure 5. Parameters of the Multinomial Observation Model Inferred from Data: **A:** Probability of counting 0, 1, ... faces at the point of fixation if the face is there, and if it's not there. **B:** Relative likelihood that a face is located N grid cells from the point of fixation, given that M face boxes were observed there. **C:** Probability of seeing M face boxes at a location N grid cells away from fixation, if the face is located there. **D:** Expected number of face boxes N grid cells away from fixation if the face is located there.

$$P(O_i = C|S = i, A = k) = \\ = \frac{T[1, |XDist(i,k)|, |YDist(i,k)|, C]}{\sum_{C'=0}^{C_{max}} T[0, |XDist(i,k)|, |YDist(i,k)|, C']} \quad (8)$$

correspond to the Bayesian MAP parameter estimates of the multinomial parameters, starting with a uniform Dirichlet conjugate prior [2].

Figure 5 shows a subset of the parameters that we fit using our entire image data set. The average number of face boxes found decreases with the face's distance to the digital fovea, showing that the face is harder to find. When there is no face, it is more likely that the face finder gives 0 face counts than if there is a face. Smaller numbers of face boxes are more likely than larger numbers regardless of whether there is a face. These results indicate that MI-POMDP is a reasonable model for object detector behavior when using a digital fovea.

4. Performance Evaluation

In the previous section, we fit the 8,820 parameters of the Multinomial detector output model to our full dataset of images. In this and following sections, all results were gathered using 7-Fold cross-validation. The images were randomly assigned to 7 groups of 500 images. In each Fold, 6 groups were used to fit the multinomial parameters, and 1 group was used to test performance. All performance results were averaged by repeating this procedure across all 7 folds. All timing experiments were done on Quad-Core Intel Xeon processors at 2.8GHz. Absolute (wall clock) time was used, with a precision of 1μs. Timing of each approach includes all the computation needed for those approaches. For MI-POMDP this includes the time needed for image cropping and downsizing, object detection, inference, and control.

4.1. Default Performance

The OpenCV 1.0 Viola-Jones Face Finding implementation has a performance parameter that controls how it searches across scales for faces. Using the default scaling parameter of 1.1, we evaluated the difference in runtime and accuracy for applying Viola Jones to a whole image, and for using Multinomial I-POMDP, which calls Viola Jones as a subroutine.

To plan fixations in a way that gathered information close to optimally, we used a policy that was shown to exhibit near-optimal fixation performance for human eyes by Butko & Movellan [4]. This policy biases fixations toward regions of the image where the face is likely to be, and once the location of the face is known with high confidence, the face is always fixated. We used a heuristic stopping criterion of the first repeated fixation. The maximum a-posteriori face location was then returned as the face location. For Viola-Jones, the grid-cell with the highest number of found face boxes was used as the face location. We measured error as the euclidean grid-cell distance from the returned face and its true location. Figure 6 shows an example of the algorithm in action. In this case, the final estimation of the face location is one grid-cell diagonal from the labeled location, giving a euclidean distance error of 1.4.

The runtime of both algorithms as a function of image size is shown in Figure 7. The runtime needed for Viola Jones is empirically linear in the number of image pixels. On our computers, it took about 1.25 ms per 1000 pixels to analyze a given image. MI-POMDP is more variable. Mostly it was linear, taking .57 ms per 1000 pixels to analyze a given image (a 2.18x speed-up). Sometimes it was very quick – much quicker than this. For a few images it was slower than Viola Jones. However, on average the real speedup (including every sub process of our algorithm) was about two-fold.

This speed increase comes at the price of a small de-

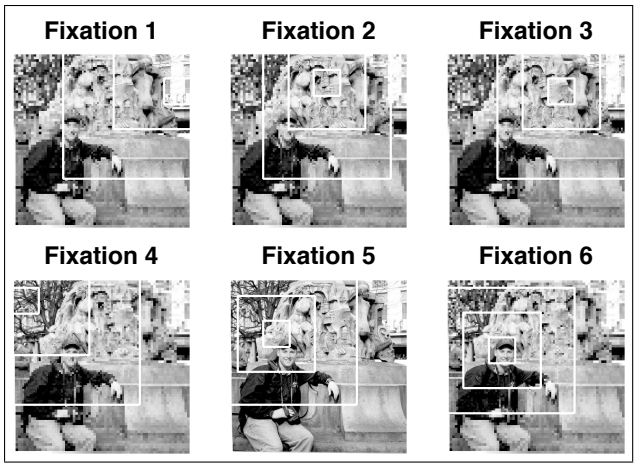

Figure 6. Successive fixation choices by the MI-POMDP policy. The face is found in six fixations. The final estimation of the face location is one grid-cell diagonal from the labeled location, giving a euclidean distance error of 1.4 grid-cells.

crease in accuracy, as shown in the Table below. Both methods on average placed the face between one and two grid-cells off the true face location.

Measure	MI-POMDP	Viola Jones
Mean Runtime (ms)	**37.9**	73.4
Scaling (ms/1000px)	**0.57**	1.25
Error (grid-cells)	1.59	**1.26**

Table 1. MI-POMDP doubles the speed of Viola-Jones with a small decrease in accuracy.

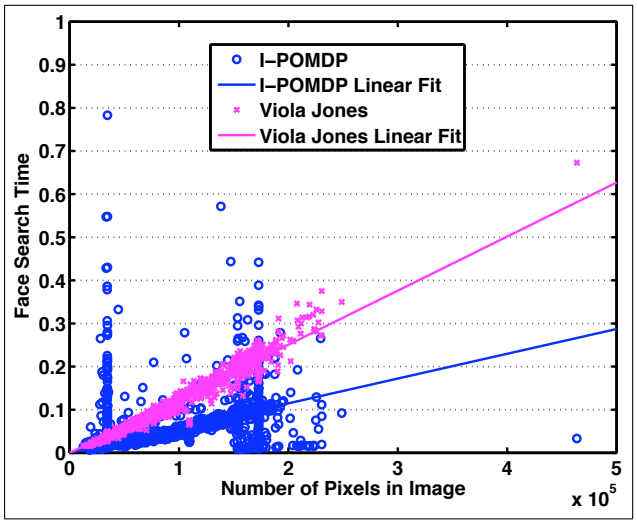

Figure 7. Time needed to search for faces, as a function of image size. A mode of the dataset image sizes was 180 × 190 (2300/3500 images), explaining apparent spike at 34,000 pixels. Similar modes explain the other spikes.

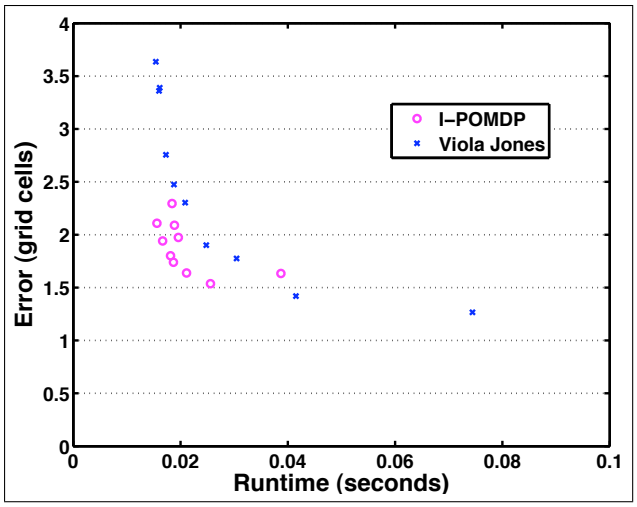

Figure 8. By changing the Viola Jones scaling factor, both Viola Jones and I-POMDP become faster and less accurate. MI-POMDP is usually closer to the origin on a time-error curve, showing that it gives a better speed-accuracy tradeoff than just applying Viola Jones.

4.2. Speed-Accuracy Tradeoff

While MI-POMDP sped up the OpenCV Face detector by a factor of two, it slightly reduced its accuracy. We thus investigated the speed-accuracy tradeoff function in OpenCV and compared it with the tradeoff provided by MI-POMDP. A speed-accuracy tradeoff function for the OpenCV classifier can be obtained by varying its scale parameter. This parameter controls the granularity of the search [15]. By default, this parameter is 1.1, but we changed it to 1.2, 1.3, ..., 2.0 and investigated the effect on speed and accuracy performance. Recall that MI-POMDP calls an object detector as a subroutine, so making that object detector faster also makes MI-POMDP faster.

Figure 8 shows that MI-POMDP on top of a Viola-Jones style object detector gives a lower runtime for a given level of error than using Viola Jones alone. Thus the MI-POMDP speed increase does not need to come with an accuracy tradeoff.

5. Conclusions and Future Work

We presented a principled model of visual search that can be used to substantially optimize the performance of generic object detectors. The approach simulates a digital fovea and scans the image so as to maximize the expected amount of information obtained about the location of the target. This is done using standard techniques from the stochastic optimal control literature. The computational cost added by this approach is more than compensated by the efficiency of the search. Speed ups of a factor of two can be expected with very little loss in accuracy. The approach proposed here

lends itself to some natural extensions:

1) We can directly optimize the policy that we use for searching, rather than relying on a policy that was shown to be near optimal for another detector. It is unknown at this point in time how much this will improve performance.

2) The approach can be integrated with saliency based search approaches, like those taken in [17]. By leveraging the Pyramid of IPs digital fovea, saliency can be computed for the foveal image representation much more quickly than for the entire image. Combined with recent fast saliency methods like [5, 3], we might expect considerable gains.

3) Digital retinas are naturally parallelizable: by simulating several fixations at once, we can gather more information more quickly. By processing all IPs at once, each fixation takes less time. A challenge will be developing optimal parallel search strategies: If you have the computational resources to search 10 locations simultaneously, which 10 would give you the best long term information gathering?

4) Extension to active cameras in robots: While a parallel implementation of Viola Jones could consider all Image Patches at once, a robot can only aim one camera at one spatial location at a time, and so it has a rigid informational bottleneck. The challenges in this extension will be in maintaining a reliable mapping from image coordinates to world coordinates, and in evaluating the foveal properties (fitting a multinomial observation model) for the robot's particular vision system.

5) More sophisticated system dynamics can be applied to search through high resolution video streams. Since the location of an object changes only a little bit frame to frame, inferences made in one frame are very informative for the next. Rather than searching the whole image for the target, we can apply one digital fixation to a frame and make inferences about where the target is (and is not) located. Since only one fixation is needed per frame, the per-image runtime will be much faster than in the current approach. While the object will not be correctly localized in every frame, once it is found, it can be easily tracked. We have already begun to explore this approach to object detection in high definition video, although at time of writing we have not quantified it thoroughly.

Acknowledgments

This work was funded by NSF Grant #ECS-0622229.

References

[1] J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, November 2001.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] N. J. Butko. Nick's Machine Perception Toolbox. `http://mplab.ucsd.edu/~nick/NMPT`, 2008.

[4] N. J. Butko and J. R. Movellan. I-POMDP: An infomax model of eye movement. In *Proceedings of the International Conference on Development and Learning (ICDL)*, August 2008.

[5] N. J. Butko, L. Zhang, G. W. Cottrell, and J. R. Movellan. Visual saliency model for robot cameras. In *International Conference on Robotics and Automation (ICRA)*, 2008.

[6] M. R. Eckhardt, I. R. Fasel, and J. R. Movellan. Towards practical facial feature detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 23, 2009.

[7] R. B. Gomes, L. M. G. Gonalves, and B. M. de Carvalho. Real time vision for robotics using a moving fovea approach with multi resolution. In *International Conference on Robotics and Automation (ICRA)*, May 2008.

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

[9] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[10] C. H. Lampert, M. B. Blaschko, and T. Hoffman. Beyond sliding windows: Object localization by efficient subwindow search. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[11] J. R. Movellan. An infomax controller for real time detection of contingency. In *Proceedings of the International Conference on Development and Learning (ICDL)*, Osaka, Japan, 2005.

[12] J. Najemnik and W. S. Geisler. Optimal eye movement strategies in visual search. *Nature*, 434:387–391, March 2005.

[13] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, 113(4):766–786, 2006.

[14] `http://mplab.ucsd.edu`. The MPLab GENKI Dataset, GENKI-SZSL Subset.

[15] `http://www.cs.indiana.edu/cgi-pub/oleykin/website/OpenCVHelp/`. The OpenCV 1.0 API.

[16] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[17] J. Vogel and N. de Freitas. Target-directed attention: Sequential decision-making for gaze planning. In *International Conference on Robotics and Automation (ICRA)*, May 2008.

[18] L. Zhang, M. H. Tong, and G. W. Cottrell. Information attracts attention: A probabilistic account of the cross-race advantage in visual search. In *Proceedings of the 29th Annual Cognitive Science Conference*, 2007.