Bayesian Optimization for Gaze Selection
CPSC 540 Course Project

Misha Denil
Department of Computer Science
University of British Columbia
Vancouver, Canada
`mdenil@cs.ubc.ca`

Abstract

This project extends the attentional tracking model developed by Bazzani et al. [1] to include gaze selection strategies which operate in the presence of partial information and on a continuous action space. We show that a straightforward extension of the existing strategy to the partial information setting results in poor performance, and we propose an alternative method based on modeling the reward surface as a Gaussian process. This approach gives good performance in the presence of partial information and allows us to expand the action space from a small, discrete set of fixation points to a continuous domain.

1 Introduction

In this project we consider tracking objects in video using a model inspired by human vision. The human visual system exploits the ability to focus attention on a narrow region of the visual field in order to cope with the vast amount of available information [2]. We mirror this structure through an appearance model with a high resolution central region and a low resolution periphery. A gaze selection strategy is learned online to choose fixation points which lead to low uncertainty in the location of the target object.

The model used in this project has been previously described in [1]. We briefly review the basic structure here, but in order to fully appreciate the setting the reader must be familiar with the original paper. Essential details of the full model which are tangential to the goals of this project, such as the design of the appearance model and update rules for the particle filter, will not be covered here.

The tracking model is composed of two interacting modules, the *what* and *where* modules, which represent the appearance of the target object and its location in the scene, respectively. This separation of responsibility is a common feature in models from the computational neuroscience literature as it is believed to reflect the structure of information processing in the human brain [3]. The *what* module compares partial observations of an object template to observations of the scene using an appearance model, and for the purposes of this project is treated as a black box process.

The *where* module is composed of a localization module and fixation module which work cooperatively to track the target. The localization module tracks the location, velocity and scale of the target using a particle filter. This component is responsible for aligning the appearance template with the full scene so that the remaining modules can operate independently of the object's position and scale. The fixation module learns a policy to select fixation points relative to an object template. These fixation points are the centers of partial template observations, and are compared with observations of the corresponding locations in the scene using the appearance model. Reward is assigned to each fixation based on the uncertainty of the target location at each time step. The fixation module uses the reward signal to adapt its gaze selection policy to achieve good localization.

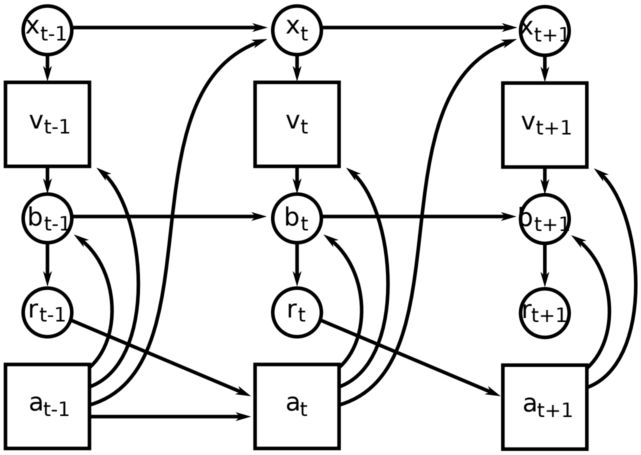

Figure 1: The full graphical model. The state of the target object is given by $\mathbf{x}_t$. This state affects the input to the appearance model $\mathbf{v}_t$, which is treated as a black box in this project. The belief state $\mathbf{b}_t$, is computed by combining the previous belief state with the current observation and the selected fixation point. This generates a reward signal, r_t which is used to update the gaze policy. At each time step the gaze selection policy is used to select a specific action $\mathbf{a}_t$. The structure of the $\mathbf{a}_t$ box depends on which gaze selection policy is used.

The graphical structure of our model is shown in Figure 1.

We consider two new implementations of the fixation module. Previous work with this model used Hedge [4, 5] to learn the gaze selection policy. We compare the performance of this algorithm to EXP3 [6, 7], which is a straightforward generalization of Hedge to the partial information setting. This is motivated by the fact that in order to build a selection policy Hedge must observe the reward not only for the selected action at each step, but also the reward which would have been received for the actions which were not chosen. Within the present framework this means evaluating all possible gaze locations at each time step, which somewhat negates the benefit of having a selection policy in the first place.

In addition to EXP3, we consider a gaze selection policy based on Bayesian optimization. Bayesian optimization is a framework for optimizing expensive cost functions [8], where in addition to finding the optimum of some objective function we have the additional constraint that we would like to find the optimum value with as few evaluations of the objective as possible. In our case we associate cost with tracking uncertainty, and our goal is to quickly find fixation points which minimize this cost.

2 Model

2.1 State-space model

The *where* module models the unobserved state of the target object $\mathbf{x}_t$, as a Markov process with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{a}_{t-1})$ where $\mathbf{a}_{t-1}$ is the fixation point chosen in the previous time step. Observations $\mathbf{v}_t$ are assumed to be conditionally independent given the current state and action. The model can be summarized as follows:

$$\begin{aligned} &p(\mathbf{x}_0) \\ &p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{a}_{t-1}) \quad \text{for } t \geq 1 \\ &p(\mathbf{v}_t|\mathbf{x}_t, \mathbf{a}_t) \quad \text{for } t \geq 1 \end{aligned}$$

We aim to estimate the filtering distribution, or *belief state*, $\mathbf{b}_t = p(\mathbf{x}_t|\mathbf{v}_{1:t}, \mathbf{a}_{1:t})$ recursively through time.[1] Since this distribution is intractable except in very simple cases, we approximate it with a particle filter [9]. The details of the particle filter are mostly tangential to this project and can be

[1] We use the notation $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$.

found in [1], what is important here is that the posterior distribution is approximated as

$$p(d\mathbf{x}_{0:t}|\mathbf{v}_{1:t},\mathbf{a}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{\tilde{\mathbf{x}}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}) \tag{1}$$

using a weighted particle population $\{\tilde{\mathbf{x}}_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^{N}$. The importance weights $w_t^{(i)}$ are used by the fixation module to determine the reward for the action chosen at time t.

2.2 Appearance model

The purpose of the appearance model is to evaluate the quality of a match between a partial observation of the object template and an observation of the scene. Object templates are formed by using optical flow to detect moving objects in the scene, and extracting a rectangular region of the image around the moving object. Comparisons between the template image and new observations are made using a factored RBM [10]. The details of how this comparison is preformed can be found in [1]. For the purposes of this project the appearance model is treated as a black box process.

2.3 Reward function

We define the instantaneous reward $r_t(\mathbf{a}_t|\mathbf{b}_t)$ as a function of the chosen action, conditioned on the belief state. We also define the cumulative reward achieved after T rounds as

$$R_T = \sum_{t=1}^{T} r_t(\mathbf{a}_t|\mathbf{b}_t) \ \ . \tag{2}$$

The goal of the gaze selection policy, is to select an action at each time step so as to maximize the cumulative reward. Many different reward functions could be used here, depending on what criteria we choose to optimize. In this project we use

$$r_t(\mathbf{a}_t|\mathbf{b}_t) = \sum_{i=1}^{N} (w_t^{(i)})^2 \ \ ,$$

which is proportional to the inverse of the effective sample size of the particle filter [11]. This choice of reward function encourages fixation points which lead to beliefs with low uncertainty.

3 Gaze control

This project compares three different strategies for learning the gaze selection policy. We use the strategy adopted in [1] as a baseline method and compare its performance to two very different alternatives.

Hedge is used in [1] to learn a randomized gaze selection policy, and the authors demonstrate that this approach performs better than other more naïve gaze selection methods. Hedge requires knowledge of the rewards for all actions at each time step, which is not realistic when gazes must be preformed sequentially, since the target object will move between fixations.

EXP3 is an extension of Hedge to partial information games [6]. Unlike Hedge, EXP3 requires knowledge of the reward only for the action selected at each time step. EXP3 is more appropriate to the setting at hand, and is also more computationally efficient than Hedge; however, this comes at a cost of substantially lower theoretical performance.

Both Hedge and EXP3 learn gaze selection policies which choose among a discrete set of predetermined fixation points. We can instead learn a continuous policy by estimating the reward surface using a Gaussian process [12]. By assuming that the reward surface is smooth, we can draw on the tools of Bayesian optimization [8] to search for the optimal gaze location using as few exploratory steps as possible.

The following sections describe the EXP3 and Bayesian optimization approaches in more detail.

3.1 EXP3

To use EXP3 [6] for gaze selection we must first discretize the action space by selecting K possible fixation points. EXP3 maintains an importance weight $\omega(i)$ for each possible fixation point and, at each time step, these weights are normalized and mixed with a uniform distribution to obtain a stochastic gaze selection policy. An action is selected according to this policy, and reward for that action is observed (this is in contrast to Hedge, which at this point must observe the rewards for each possible action). The observed reward is then used to update the importance weights and the process repeats. Pseudo code for EXP3 is shown in Algorithm 1.

Algorithm 1 EXP3

```
Input: $\gamma \in (0,1]$
Input: $\omega_1(i) = 1$ for $i \in$ Actions
  for $t = 1, 2, \ldots$ do
    $p_t(i) \leftarrow (1-\gamma)\frac{\omega_t(i)}{\sum_j \omega_t(j)} + \frac{\gamma}{K}$
    $\mathbf{a}_t \sim (p_t(1), \ldots, p_t(K))$        // sample from the distribution $(p_t(i))$
    $r_t(i) \leftarrow r_t(\mathbf{a}_t|\mathbf{b}_t, \boldsymbol{\theta}_t)$
    for $j \in$ Actions do
      $\hat{r}_t(j) \leftarrow \begin{cases} r_t(j)/p_t(j) & \text{if} j = \mathbf{a}_t \\ 0 & \text{otherwise} \end{cases}$
      $\omega_{t+1}(j) \leftarrow \omega_t(j)\exp\left(\gamma \hat{r}_t(j)/K\right)$
    end for
  end for
```

3.2 Bayesian optimization

Both Hedge and EXP3 rely on discretizing the space of possible fixation points and learn a distribution over this finite set. In contrast, Bayesian optimization is able to treat the space of possible fixation points as fully continuous by placing a smoothness prior on how reward is expected to vary with respect to location. Intuitively, if we know the reward at one location, then we expect other, nearby locations to produce similar rewards. Gaussian process priors encode this type of belief [12], and have been used extensively for optimization of cost functions when it is important to minimize the total number of function evaluations [8].

We model the reward function $r_t(\mathbf{a}_t|\mathbf{b}_t) \triangleq r(\mathbf{a}_t|\mathbf{b}_t, \boldsymbol{\theta}_t)$ as a zero mean Gaussian process

$$r(\mathbf{a}_t|\mathbf{b}_t, \boldsymbol{\theta}_t) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{a}_t, \mathbf{a}_t'|\mathbf{b}_t, \boldsymbol{\theta}_t)) \ \ ,$$

where $\mathbf{b}_t$ is the belief state, (see Section 2), and $\boldsymbol{\theta}_t$ are the model hyperparameters (see Section 3.3). The kernel function $k(\cdot, \cdot)$, gives the covariance between the reward at any two gaze locations. For notational simplicity the explicit dependence of $r(\cdot)$ and $k(\cdot,\cdot)$ on $\mathbf{b}_t$ and $\boldsymbol{\theta}_t$ will be dropped.

Given a set of observations we can compute the posterior predictive distribution for $r(\cdot)$

$$\begin{aligned} r(\mathbf{a}|\mathbf{r}_{1:t}, \mathbf{a}_{1:t}) &\sim \mathcal{N}(m_t(\mathbf{a}), s_t^2(\mathbf{a})) \\ m_t(\mathbf{a}) &= \mathbf{k}^{\mathrm{T}}[\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1}\mathbf{r}_{1:t} \\ s_t^2(\mathbf{a}) &= k(\mathbf{a}, \mathbf{a}) - \mathbf{k}^{\mathrm{T}}[\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1}\mathbf{k} \end{aligned} \tag{3}$$

where σ_n^2 is a hyperparameter indicating the level of noise in our observations which we absorb into $\boldsymbol{\theta}_t$, and

$$\begin{aligned} \mathbf{K} &= \begin{bmatrix} k(\mathbf{a}_1, \mathbf{a}_1) & \cdots & k(\mathbf{a}_1, \mathbf{a}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{a}_t, \mathbf{a}_1) & \cdots & k(\mathbf{a}_t, \mathbf{a}_t) \end{bmatrix} \\ \mathbf{k} &= \begin{bmatrix} k(\mathbf{a}_1, \mathbf{a}) & \cdots & k(\mathbf{a}_t, \mathbf{a}) \end{bmatrix}^{\mathrm{T}} \\ \mathbf{r}_{1:t} &= \begin{bmatrix} r_1 & \cdots & r_t \end{bmatrix}^{\mathrm{T}} \ \ . \end{aligned}$$

Equation 3 is a Gaussian process estimate of the reward surface and can be used to select a fixation point for the next time step. This estimate gives both a predicted reward value and an associated uncertainty for each possible fixation point. This is the strength of Gaussian processes for this type of optimization problem, since the predictions can be used to balance exploration (choosing a fixation point where the reward is highly uncertain) and exploitation (choosing a point we are confident will have high reward).

There are many selection methods available in the literature which offer different tradeoffs between these two criteria. In this project we use GP-UCB [13] which selects

$$\mathbf{a}_{t+1} = \arg\max_{\mathbf{a}} m_t(\mathbf{a}) + \sqrt{\beta_t} s_t(\mathbf{a}) \tag{4}$$

where β_t is a parameter. The setting $\beta_t = 2\log(t^3\pi^2/3\delta)$ (with $\delta = 0.001$) is used throughout this project, since it has been shown that the cumulative regret (i.e. the gap between Equation 2 and the optimal R_T) with this parameter setting grows sub-linearly with time with overwhelming probability.

Equation 4 must still be optimized to find $\mathbf{a}_{t+1}$, which can be performed using standard global optimization tools. We use `DIRECT` [14] in this project due to the existence of a readily available implementation.

3.3 Selecting the kernel function

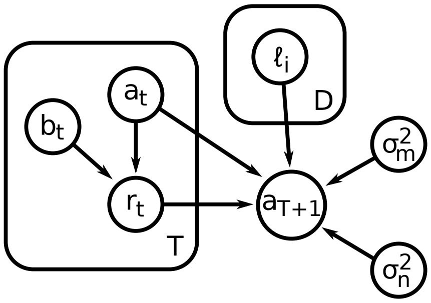

Figure 2: Graphical model for Bayesian optimization. The ℓ_i are length scales in each dimension, σ_m^2 is the magnitude parameter and σ_n^2 is the noise level. In our model σ_m^2 and σ_n^2 follow a uniform prior and the ℓ_i follow independent student-t priors.

We consider three different possibilities for the Gaussian process kernel function. The first is the squared exponential kernel, which computes the covariance between different actions as

$$k(\mathbf{a}_i, \mathbf{a}_j) = \sigma_m^2 \exp\left(-\frac{1}{2}r^2\right) \quad . \tag{5}$$

We also consider two kernels from the Mátern family. Kernels in this family are indexed by a parameter ν, and have the general form

$$k(\mathbf{a}_i, \mathbf{a}_j) = \sigma_m^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu}r)^\nu K_\nu(\sqrt{2\nu}r) \quad , \tag{6}$$

where $K_\nu(\cdot)$ is a modified Bessel function of the second kind. The Mátern kernels have a simple closed form for half integer values of ν [12], and we consider specifically the cases where $\nu = 3/2$ and $\nu = 5/2$. In Equations 5 and 6, the value of r is given by

$$r^2 = \sum_{k=1}^{D} \left(\frac{a_{i,k} - a_{j,k}}{\ell_k}\right)^2 \quad ,$$

where the summation runs over the dimensions of the action space ($D = 2$ in our case).

The GP regression is controlled by several hyperparameters: σ_m^2 controls the overall magnitude of the covariance, and σ_n^2 (see Equation 3) controls the amount of observation noise. The remaining parameters $\{\ell_1, \ldots, \ell_D\}$ are length scale parameters which control the range of the covariance effects in each dimension.

Treatment of the hyperparameters requires special consideration in this setting. The pure Bayesian approach is to put a prior on each parameter and integrate them out of the predictive distribution. However, since the integrals involved are not tractable analytically, this requires computationally expensive numerical approximations. Speed is an issue here since GP-UCB requires that we optimize a function of the posterior process at each time step so, for instance, computing Monte Carlo averages for each evaluation of Equation 3 is prohibitively slow.

An alternative approach is to choose parameter values via maximum likelihood. This can be done very quickly, and allows us to make speedy predictions; however, in this case we suffer from problems of data scarcity, particularly early in the tracking process when few observations have been made. The length scale parameters are particularly prone to receiving very poor estimates when there is little data available.

We have found that using informative priors for the length scale parameters and making MAP, rather than ML, estimates at each time step provides a solution to the problems described above. MAP estimates can be made quickly using gradient optimization methods [12], and informative priors provide resistance to the problems encountered with ML. The experiments in Section 4 place uniform priors on the magnitude and noise parameters and place independent Student-t priors on each length scale parameter. The experiments also use an initial data collection phase of 10 time steps before any adjustment of the parameters is made.

4 Experiments

In this section, three experiments are carried out to evaluate the performance of the different gaze selection policies. The first experiment examines how the performance of Bayesian optimization varies with respect to the selection of kernel function, and the degree of smoothness and informativeness of the length scale prior. We demonstrate that the performance of of Bayesian optimization is not strongly affected by the choice of kernel or prior. The data set for this experiment consists of several videos of digits from the MNIST data set moving on a black background. The target in each video encounters one or more partial occlusions which the tracking algorithm must handle gracefully.

In the second experiment we compare the performance of each gaze selection method on the same videos except this time each sequence has been corrupted by 30% noise. We measure the error between the estimated track and the ground truth for each gaze selection method, and demonstrate that Bayesian optimization preforms comparably to Hedge, but that EXP3 is not able to reach a satisfactory level of performance. We also demonstrate qualitatively that the Bayesian optimization approach learns good gaze selection policies on this data set.

Finally, our third experiment provides evidence that the Bayesian optimization method can generalize to real world data.

The results of our first experiment are shown in Figure 3. The success rate is measured as the proportion of runs where the algorithm is able to track the target throughout the entire test sequence. The success rate is computed for each digit across ten runs and the results in Figure 3 show distributions over digit classes at each scale. The success rate varies across the various permutations of

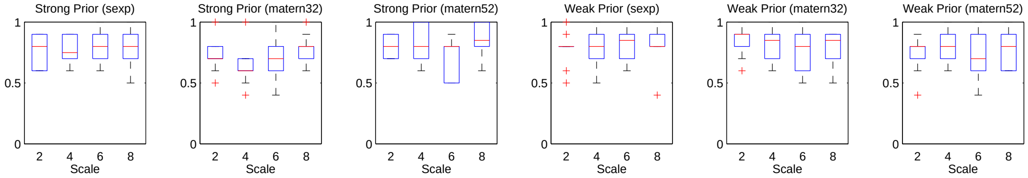

Figure 3: Results from the first experiment. On the left are success rates for each kernel type using a strong prior for the length scale parameters. On the right is results for the same kernels using weaker priors on the length scale. In all cases the noise and magnitude priors are uniform. The length scale parameters have Student-t priors with the mean indicated along the x-axis. The strong priors use a scale of 3 with 3 degrees of freedom and the weak priors use a scale of 10 with 1 degree of freedom.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Avg**
Bayesopt	5.36 (2.32)	7.92 (2.52)	2.62 (3.89)	4.05 (1.67)	1.70 (5.10)	8.31 (3.35)	4.94 (2.28)	12.09 (3.53)	1.52 (2.76)	9.06 (1.66)	**5.76** **(2.91)**
Hedge	2.97 (1.56)	3.20 (2.19)	2.97 (1.99)	2.92 (2.00)	3.14 (1.80)	2.96 (2.08)	2.86 (1.96)	2.98 (1.76)	2.81 (1.64)	3.15 (3.73)	**3.00** **(2.07)**
EXP3	3.18 (5.05)	3.03 (10.08)	65.46 (3212.16)	91.81 (3671.66)	2.62 (2.35)	7.20 (303.29)	67.54 (2346.82)	2.97 (3.99)	3.06 (2.71)	77.01 (3135.17)	**32.39** **(1269.33)**

Table 1: Tracking error on several video sequences using different methods for gaze selection. The table shows mean tracking error as well as the error variance (in brackets) over a single test sequence.

priors and kernel types; however, this variation is small and the performance is consistently high. This indicates that the choice of kernel and prior are not essential to the performance of Bayesian optimization on this task, and following this we select the squared exponential kernel with a weak prior and a length scale of 8 to use for the remaining experiments.

Table 1 reports the results from our second experiment. The table shows the mean tracking error, measured by averaging distance between the estimated and ground truth track over the entire video sequence. Here we see that the Bayesian optimization approach compares favorably to Hedge in terms of tracking performance, and that EXP3 preforms substantially worse than the other two methods. Although Hedge preforms marginally better than Bayesian optimization, it is important to remember that Bayesian optimization solves a significantly more difficult problem. Hedge relies on discretizing the action space, and must have access to the rewards for all possible actions at each time step. In contrast, Bayesian optimization considers a fully continuous action space, and receives reward information only for the chosen actions.

Figure 4 shows the reward surfaces learned for each digit by Bayesian optimization, as well as a visualization of the overall best fixation points using data aggregated across ten runs. The optimal fixation points found by the algorithm are tightly clustered, and the resulting observations are very distinguishable.

In our third experiment we use the Youtube celebrity dataset from [15]. This data set consists of several videos of celebrities taken from Youtube and is challenging for tracking algorithms as the videos exhibit a wide variety of illuminations, expressions and face orientations. We run our tracking model using Bayesian optimization to learn a gaze selection policy on this data set, and present some results in Figure 5. Although we report only qualitative results from this experiment, it provides anecdotal evidence that Bayesian optimization is able to form a good gaze selection policy on real world data.

5 Conclusions and future work

In this project we considered two new gaze selection policies for the visual tracking model proposed in [1]. The first is a straightforward extension of the existing policy to the partial information case; however, we saw that this method gave poor tracking performance even on a relatively simple synthetic problem.

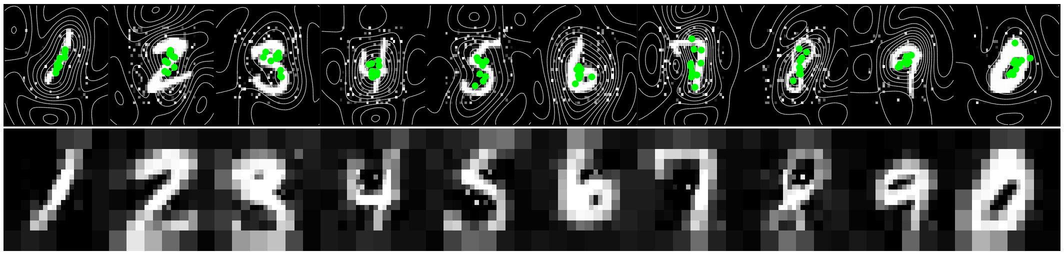

Figure 4: **Top:** Digit templates with the estimated reward surfaces superimposed. Markers indicate the best fixation point found in each of ten runs. **Bottom:** A visualization of the image found by averaging the best fixation points found across ten runs.

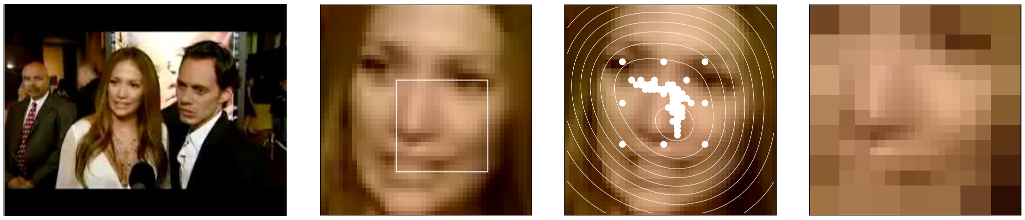

Figure 5: Results on a real data set. **Far left:** An example frame from the video sequence. **Center left:** The tracking template with the optimal fixation window highlighted. **Center right:** The reward surface produced by Bayesian optimization. The white markers show the centers of each fixation point in a single tracking run. **Right:** Input to the observation model when fixating on the best point. (Best viewed from a distance).

The second method we considered, based on Bayesian optimization, is able not only to perform well in the presence of partial information but also allows us to expand the set of possible fixation points to a continuous domain. We saw that this approach performs comparably to the original method on synthetic data and that it is able to generalize to track objects in real world videos.

There are several possible extensions to this line of work. For instance, the current model has no ability to recover from a tracking failure. In [1] it was shown how to classify the target as it is being tracked, and it may be possible to use this classification information to detect and recover from tracking failure.

The gaze selection policies considered here assume that the reward surface is static throughout time; however, changing appearance of the target may cause the optimal fixation point to shift.

A closer examination of the exploration/exploitation tradeoff in the tracking setting is in order. For instance, the methods we considered assume that future rewards are independent of past actions. This assumption is clearly not true in our setting, since choosing a long sequence of very poor fixation points can lead to tracking failure. The reinforcement learning community has long considered models which allow control over the relative importance of near-term vs long-term rewards, and it may be possible to incorporate these techniques into the present framework. Other more direct approaches which control exploration as a function of tracking uncertainty may also be effective.

References

[1] L. Bazzani, N. de Freitas, and J.A. Ting. Learning attentional mechanisms for simultaneous object tracking and recognition with deep networks. *NIPS 2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010.

[2] R.A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7(1):17–42, 2000.

[3] B.A. Olshausen, C.H. Anderson, and D.C. Van Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *The Journal of Neuroscience*, 13(11):4700, 1993.

[4] P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *focs*, page 322. Published by the IEEE Computer Society, 1998.

[5] Y. Freund. Schapire, A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 95:119–139, 1997.

[6] P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2001.

[7] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[8] E. Brochu, V.M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical report, University of British Columbia, 2010.

[9] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[10] A.K. MarcAurelio Ranzato and G.E. Hinton. Factored 3-way restricted boltzmann machines for modeling natural images. In *Artificial Intelligence and Statistics*, 2010.

[11] J.S. Liu. *Monte Carlo strategies in scientific computing*. Springer Verlag, 2008.

[12] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

[13] N. Srinivas, A. Krause, S.M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning*, 2010.

[14] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[15] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. *IEEE Conf. Computer Vision and Pattern Recognition*, 2008.