Learning attentional mechanisms for simultaneous object tracking and recognition with deep networks

Loris Bazzani
Computer Science
U. of Verona
Verona, Italy
loris.bazzani@univr.it

Nando de Freitas
Computer Science
U. of British Columbia
Vancouver, Canada
nando@cs.ubc.ca

Jo-Anne Ting
Computer Science
U. of British Columbia
Vancouver, Canada
jating@cs.ubc.ca

Abstract

We propose a novel attentional model for simultaneous object tracking and recognition that is driven by gaze data. Motivated by theories of the human perceptual system, the model consists of two interacting pathways: ventral and dorsal. The ventral pathway models object appearance and classification using deep (factored)-restricted Boltzmann machines. At each point in time, the observations consist of retinal images; with decaying resolution toward the periphery of the gaze. The dorsal pathway models the location, orientation, scale and speed of the attended object. The posterior distribution of these states is estimated with particle filtering. Deeper in the dorsal pathway, we encounter an attentional mechanism that learns to control gazes so as to maximize different objectives. Here we demonstrate the method when the objective is to minimize the uncertainty in the posterior distribution of the states. The approach is modular (with each module easily replaceable with more sophisticated algorithms), straightforward to implement, practically efficient, and works well in simple video sequences.

1 Introduction

Humans track and recognize objects effortlessly and efficiently, exploiting attentional mechanisms [24, 6] to cope with a vast stream of data. In this paper, we use the human visual system as inspiration to build a model for simultaneous object tracking and recognition from gaze data, as shown in Figure 1. The proposed model also addresses the problem of gaze planning (i.e., where to look in order to achieve some goal, such as minimizing position, speed or object label uncertainty).

The model consists of two interacting modules, ventral and dorsal, which are also known as the *what* and *where* modules respectively. The dorsal pathway is in charge of state estimation and control. At the lowest level, a particle filter [7] is used to estimate the states of the object under consideration, including location, orientation, speed and scale. We make no attempt to implement such states with neural architectures, but it seems clear that they could be encoded with grid cells [18] and retinotopic maps as in V1 and the superior colliculus [25, 9]. At the higher level of the dorsal pathway, a policy governing where to gaze next is learned with an online hedging algorithm [1]. This policy learning step could be easily improved using other bandit approaches and Bayesian optimization [3, 4, 5]. The dorsal attentional mechanism is responsible for controlling saccades and, to a significant extent, smooth pursuit [6].

The ventral pathway consists of two layers as shown in Figure 1; one for learning latent representations and a second for object recognition. We use (factored)-restricted Boltzmann machines (RBMs) [11, 23, 29], but autoencoders [28], sparse coding [21, 14], two-layer ICA [15] and convolutional architectures [17] could also be adopted in this module. At present, we pre-train the appearance models.

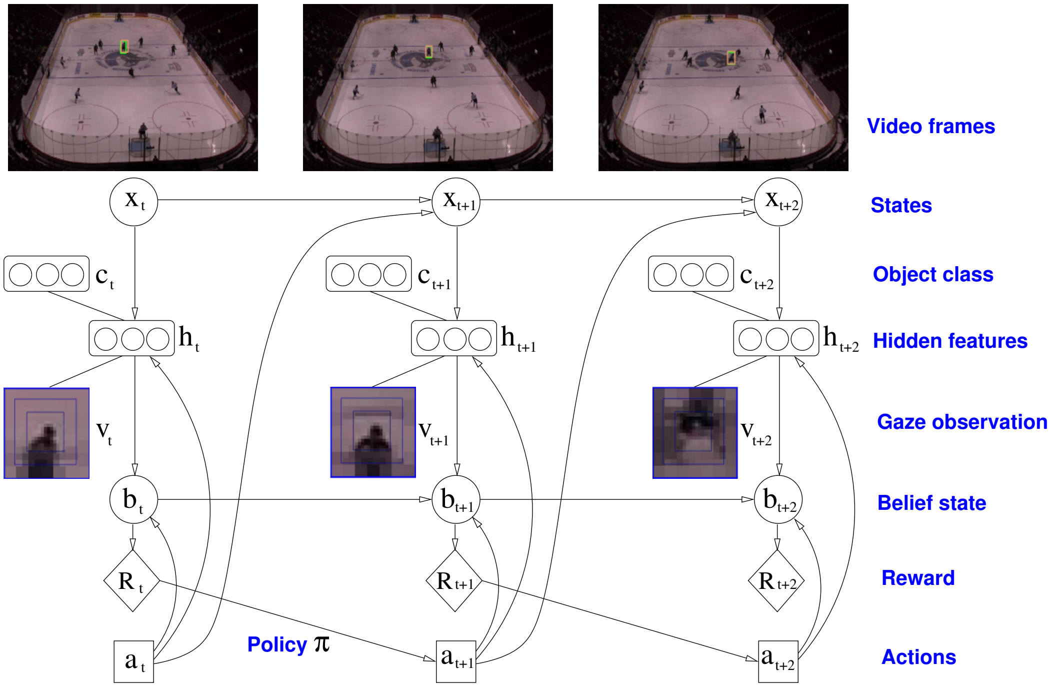

Figure 1: *Graphical model. From the gaze data $\mathbf{v}_t$, the model infers the hidden features $\mathbf{h}_t$ (that is, the activation intensity of each receptive field) and the object class $\mathbf{c}_t$ at time t. The location, size, speed and orientation of the gaze patch is determined by the state $\mathbf{x}_t$. The actions $\mathbf{a}_t$ follow a randomized policy π_t that depends on the cumulative reward R_{t-1}. This particular reward is a function of the belief state $\mathbf{b}_t = p(\mathbf{x}_t|\mathbf{a}_{1:t}, \mathbf{h}_{1:t})$, also known as the filtering distribution. Unlike typical commonly used partially observed Markov decision models (POMDPs), the reward is a function of the beliefs. In this sense, the problem is closer to one of sequential experimental design. Alternative reward and policy models, aimed at improving classification, could be easily obtained by adding directed edges from $\mathbf{c}_t$ to $\mathbf{a}_t$ and R_t. With more layers in the ventral $\mathbf{v} - \mathbf{h} - \mathbf{c}$ pathway, other rewards and policies could be designed to implement higher-level attentional strategies.*

The proposed system can be motivated from different perspectives. First, starting with [12], many particle filters have been proposed for image tracking, but these typically use simple observation models such as B-splines [12] and color templates [19]. RBMs are more expressive models of shape, and hence, we conjecture that they will play a useful role where simple appearance models fail. Second, from a deep learning computational perspective, this work allows us to tackle large images and video. The use of fixations synchronized with information about the state (e.g. location and scale) of such fixations, eliminates the need to look at the entire image or video. Third, the system is invariant to image transformations encoded in the state, such as location, scale and orientation. Fourth, from a dynamic sensor network perspective, this paper presents a very simple, but efficient, novel way of deciding how to gather measurements dynamically. Lastly, in the context of psychology, the proposed model realizes to some extent the functional architecture for dynamic scene representation of [24]. The rate at which different attentional mechanisms develop in newborns (including alertness, saccades and smooth pursuit, attention to object features and high-level task driven attention) guided the design of the proposed approach and was a great source of inspiration [6].

Recently, a dynamic RBM state space model was proposed in [27]. Both the implementation and intention behind that proposal are different from the approach discussed here. To the best of our knowledge, the approach presented here is the first successful attempt to combine dynamic state estimation from gazes with online policy learning for gaze adaptation, using deep belief network models of appearance. Many other dual-pathway architectures have been proposed in computational neuroscience, including [20, 22], but we believe ours has the advantage that it is very simple, *modular* (with each module easily replaceable), easy to implement, suitable for large datasets and easy to extend.

2 Model specification

2.1 State-space model

The standard approach to image tracking is based on the formulation of Markovian, nonlinear, non-Gaussian state-space models, which are solved with approximate Bayesian filtering techniques. In this setting, the unobserved signal (object's position, velocity, scale, orientation or discrete set of operation) is denoted $\{\mathbf{x}_t \in \mathcal{X}; t \in \mathbb{N}\}$. This signal has initial distribution $p(\mathbf{x}_0)$ and transition equation $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1})$[1]. Here $\mathbf{a}_t \in \mathcal{A}$ denotes an action, at time t, defined on a discrete state space of size $|\mathcal{A}| = K$. The observations $\{\mathbf{h}_t \in \mathcal{H}; t \in \mathbb{N}^*\}$, are assumed to be conditionally independent given the state process $\{\mathbf{x}_t; t \in \mathbb{N}\}$. In summary, the state-space model is described by the following distributions:

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

where $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, ..., \mathbf{x}_t\}$ and $\mathbf{h}_{1:t} \triangleq \{\mathbf{h}_1, ..., \mathbf{h}_t\}$ represent the states and the observations up to time t, respectively. For the transition model, we will adopt a classical autoregressive process. The appearance model $p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t)$ is slightly more involved and will be discussed in Section 2.3.

Our aim is to estimate recursively in time the *posterior distribution* $p(\mathbf{x}_{0:t} | \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ and its associated features, including the marginal distribution $p(\mathbf{x}_t | \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ — known as the *filtering distribution* or *belief state*. This distribution satisfies the following recurrence:

$$\mathbf{b}_t \triangleq p(\mathbf{x}_t|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) \propto p(\mathbf{h}_t|\mathbf{x}_t, \mathbf{a}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{a}_{t-1}) p(d\mathbf{x}_{t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}).$$

Except for standard distributions (*e.g.* Gaussian or discrete), this recurrence is intractable.

2.2 Reward function and policy

To complete the specification of the model, we need to introduce the policy $\pi(\cdot)$ and an instantaneous reward function $r_t(\cdot)$. The reward can be any desired behavior for the system, such as maximizing classification accuracy, minimizing posterior uncertainty, or achieving a more abstract goal. To ground the discussion, however, we focus on gathering observations so as to minimize the uncertainty in the estimate of the filtering distribution: $r_t(\mathbf{b}_t) \triangleq u[\widetilde{p}(\mathbf{x}_t|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})]$. More specifically, as discussed later, this reward will correspond to the variance of the importance weights of the particle filter approximation $\widetilde{p}(\mathbf{x}_t|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ of the belief state. In our current implementation, each action is a different gaze location. The objective is to choose where to look so as to minimize the uncertainty about the belief state.

We also need to introduce the cumulative reward of the control algorithm for each action:

$$R_T(\mathbf{a}_T = k) = \sum_{t=1}^{T} r_t(p(\mathbf{x}_t|\mathbf{h}_{1:t}, \mathbf{a}_t = k, \mathbf{a}_{1:t-1})).$$

The actions are distributed according to the following stochastic policy:

$$\pi_t(\mathbf{a}_t = k|R_{t-1}) = \frac{\exp(\eta R_{t-1}(\mathbf{a}_t = k))}{\sum_{j=1}^{K} \exp(\eta R_{t-1}(\mathbf{a}_t = j))},$$

where $\eta > 0$ is a parameter. We have defined the policy in this way as it enables us to borrow decision making algorithms from the online learning framework [1] with very little effort. Here, we will adopt the *hedge algorithm* for full information games described in [1]. (This algorithm has vanishing regret.) However, as mentioned in the introduction, one could adopt other bandit techniques [4, 5], Bayesian optimization [3] or mirror descent [2] to extend the policy to continuous action spaces and treat imperfect information games.

[1] For simplicity, we use $\mathbf{x}_t$ to denote both the random variable and its realization, unless we feel it is necessary to make this distinction explicit. Consequently, we express continuous probability distributions using $p(d\mathbf{x}_t)$ instead of $\Pr(\mathbf{X}_t \in d\mathbf{x}_t)$ and discrete distributions using $p(\mathbf{x}_t)$ instead of $\Pr(\mathbf{X}_t = \mathbf{x}_t)$. If these distributions admit densities with respect to an underlying measure μ (usually counting or Lebesgue), we denote these densities by $p(\mathbf{x}_t)$. To make the material accessible to a wider audience, we shall allow for a slight abuse of terminology by, sometimes, referring to $p(\mathbf{x}_t)$ as a distribution.

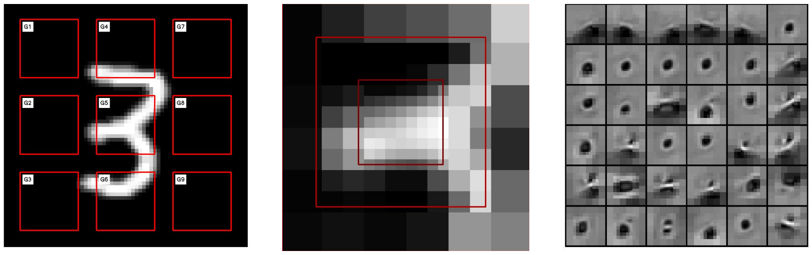

Figure 2: *(a) Template with 9 gazes initialized automatically when motion is detected. (b) Foveal observation corresponding to gaze G5 in the template. (c) The most active RBM filters for this observation.*

2.3 Appearance model

We use (factored)-RBMs to model the appearance of objects and perform object classification using the gazes chosen by the control module. These undirected probabilistic graphical models are governed by a Boltzmann distribution $p(\mathbf{v}, \mathbf{h}|\mathbf{w})$ over the gaze data $\mathbf{v}_t$ and the hidden features $\mathbf{h}_t \in \{0,1\}^{n_h}$. We assume that the receptive fields $\mathbf{w}$, also known as RBM weights or filters, have been trained beforehand. We also assume that readers are familiar with these models and, if otherwise, refer them to [23, 26].

In image tracking, the observation model is often defined in terms of the distance of the observations with respect to a template τ,

$$p\left(\mathbf{h}_t|\mathbf{x}_t, \mathbf{a}_t\right) \propto e^{-d(\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t), \tau)},$$

where $d(\cdot,\cdot)$ denotes a distance metric and τ an object template (for example, a color histogram or spline). In this model, the observation $\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t)$ is a function of the current state hypothesis and the selected action. The problem with this approach is eliciting a good template. Often color histograms or splines are insufficient. For this reason, we will construct the templates with (factored)-RBMs as follows. First, optical flow is used to detect new object candidates entering the visual scene. Second, we assign a template to the detected object candidate, which consists of several gazes covering the field of motion, as shown in Figure 2 for $K = 9$ gazes. The same figure also shows the typical foveated observations (higher resolution in the center and lower in the periphery of the gaze) and the receptive fields for these observations learned beforehand with an RBM. The control algorithm will be used to learn which of the gazes in the template are more fruitful. That is, each "saccadic" action will correspond to the selection of one of these gaze options. Finally, we define the likelihood of each observation directly in terms of the distance of the hidden units of the RBM $\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t, \mathbf{v}_t)$ to the hidden units of each template region $\mathbf{h}(\mathbf{x}_1, \mathbf{a}_t = k, \mathbf{v}_1)$, $k = 1 : K$, initialized in the first frame. That is,

$$p\left(\mathbf{h}_t|\mathbf{x}_t, \mathbf{a}_t = k\right) \propto e^{-d(\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t = k, \mathbf{v}_t), \mathbf{h}(\mathbf{x}_1, \mathbf{a}_t = k, \mathbf{v}_1))}.$$

The above template is static, but conceivably one could adapt it over time.

The appearance module also performs object recognition, classifying a gaze instance selected with the gaze policy. The input of the classifier is the latent representation $\mathbf{h}_t$. To estimate the class variable $\mathbf{c}_t$ (Figure 1) over time, we accumulate the classification decisions at each time step. This simple classifier ignores temporal coherence between gazes. A better strategy would be to perform classification on a sequence of gazes, as recently demonstrated in [16].

3 Algorithm

Since the belief state cannot be computed analytically, we will adopt particle filtering to approximate it. The algorithm is shown in Figure 3. We refer readers to [7] for a more in depth treatment of these sequential Monte Carlo methods. Assume that at time $t-1$ we have $N \gg 1$ particles (samples) $\{\mathbf{x}_{0:t-1}^{(i)}\}_{i=1}^{N}$ distributed according to $p\left(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}\right)$. We can approximate this belief state with the following empirical distribution $\widehat{p}\left(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}\right) \triangleq \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_{0:t-1}^{(i)}}\left(d\mathbf{x}_{0:t-1}\right)$. Particle filters combine sequential importance sampling with a selec-

```
1. Initialization, $t = 0$.
    • For $i = 1, ..., N$, sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, set $t = 1$, and initialize the template.
    • Set $R_0(\mathbf{a}_0 = k) = 0$ for $k = 1, \ldots, K$, where $K$ is the number of gazes in the template
2. Importance sampling step
    • For $i = 1, ..., N$, sample $\widetilde{\mathbf{x}}_t^{(i)} \sim q_t\left(d\mathbf{x}_t^{(i)} \middle| \widetilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}\right)$ and set $\widetilde{\mathbf{x}}_{0:t}^{(i)} = \left(\mathbf{x}_{0:t-1}^{(i)}, \widetilde{\mathbf{x}}_t^{(i)}\right)$.
    • For $i = 1, ..., N$, $k = 1, \ldots, K$, , evaluate the importance weights
          $\widetilde{w}_t^{(i),k} \propto \frac{p\left(\mathbf{h}_t | \widetilde{\mathbf{x}}_t^{(i)}, \mathbf{a}_t = k\right) p\left(\widetilde{\mathbf{x}}_t^{(i)} | \widetilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{a}_{t-1}\right)}{q_t\left(\widetilde{\mathbf{x}}_t^{(i)} \middle| \widetilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}\right)}.$
    • Normalise the importance weights $w_t^{(i),k} = \frac{\widetilde{w}_t^{(i),k}}{\sum_{j=1}^{N} \widetilde{w}_t^{(j),k}}$
3. Gaze control step
    • Update the policy
          $\pi_t(a_t = k | R_{t-1}) = \frac{\exp(\eta R_{t-1}(\mathbf{a}_t = k))}{\sum_{j=1}^{K} \exp(\eta R_{t-1}(\mathbf{a}_t = j))}$
    • Receive the rewards $r_{t,k} = \sum_{i=1}^{N} \left(w_t^{(i),k}\right)^2$ for $k = 1, \ldots, K$
    • Set $R_t(\mathbf{a}_t = k) = R_{t-1}(\mathbf{a}_t = k) + r_{t,k}$ for $k = 1, \ldots, K$
    • Sample action $k^\star$ according to the policy $\pi_t(\cdot)$.
    • Set $w_t^{(i)} = w_t^{(i),k^\star}$ for $i = 1, ..., N$
4. Selection step
    • Resample with replacement $N$ particles $\left(\mathbf{x}_{0:t}^{(i)}; i = 1, \ldots, N\right)$ from the set $\left(\widetilde{\mathbf{x}}_{0:t}^{(i)}; i = 1, \ldots, N\right)$ according to the normalized importance weights $w_t^{(i)}$.
    • Set $t \leftarrow t + 1$ and go to step $2$.
```

Figure 3: *Particle filtering algorithm with gaze control.*

tion scheme designed to obtain N new particles $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^{N}$ distributed approximately according to $p(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$.

3.1 Importance sampling step

The joint distributions $p(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1})$ and $p(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ are of different dimension. We first modify and extend the current paths $\mathbf{x}_{0:t-1}^{(i)}$ to obtain new paths $\widetilde{\mathbf{x}}_{0:t}^{(i)}$ using a proposal kernel $q_t(d\widetilde{\mathbf{x}}_{0:t}|\mathbf{x}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$. As our goal is to design a sequential procedure, we set $q_t(d\widetilde{\mathbf{x}}_{0:t}|\, \mathbf{x}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = \delta_{\mathbf{x}_{0:t-1}}(d\widetilde{\mathbf{x}}_{0:t-1})\, q_t(d\widetilde{\mathbf{x}}_t|\, \widetilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$, that is $\widetilde{\mathbf{x}}_{0:t} = (\mathbf{x}_{0:t-1}, \widetilde{\mathbf{x}}_t)$. The aim of this kernel is to obtain new paths whose distribution $q_t(d\widetilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = p(d\widetilde{\mathbf{x}}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1})\, q_t(d\widetilde{\mathbf{x}}_t|\, \widetilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ is as "close" as possible to $p(d\widetilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$. Since we cannot choose $q_t(d\widetilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = p(d\widetilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ because this is the quantity we are trying to approximate in the first place, it is necessary to weight the new particles so as to obtain consistent estimates. We perform this "correction" with importance sampling, using the weights:

$$\widetilde{w}_t = \frac{p(d\widetilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})}{q_t(d\widetilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})} = \widetilde{w}_{t-1} \frac{p(\mathbf{h}_t|\widetilde{\mathbf{x}}_t, \mathbf{a}_t)\, p(d\widetilde{\mathbf{x}}_t|\widetilde{\mathbf{x}}_{0:t-1}, \mathbf{a}_{t-1})}{q_t(d\widetilde{\mathbf{x}}_t|\, \widetilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})}.$$

The choice of the transition prior as proposal distribution is by far the most common one. In this case, the importance weights reduce to the expression for the likelihood. However, it is possible to construct better proposal distributions, which make use of more recent observations, using object detectors [19], saliency maps [13], optical flow, and approximate filtering methods, as in the unscented particle filter. One could also incorporate strategies to manage data association and other tracking related issues. We obviate these issues to focus instead on components of this project that are more relevant to the deep learning community.

After normalizing the weights, $w_t^{(i)} = \frac{\widetilde{w}_t^{(i)}}{\sum_{j=1}^N \widetilde{w}_t^{(j)}}$, we obtain the following estimate of the filtering distribution:

$$\widetilde{p}\left(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t},\mathbf{a}_{1:t}\right) = \sum_{i=1}^{N} w_t^{(i)} \delta_{\widetilde{\mathbf{x}}_{0:t}^{(i)}}\left(d\mathbf{x}_{0:t}\right).$$

3.2 Gaze control step

We treat the problem of choosing a gaze (from the template) with a portfolio allocation algorithm called Hedge [8, 1]. Hedge is an algorithm that, at each time step t, updates the policy $\pi_t(\mathbf{a}_t|R_{t-1})$ for each allowed action (see [1]). It then selects an action $k^\star$ according to this policy as shown in Figure 3. The immediate reward is defined as the variance of the normalized importance weights. This choice is motivated by the fact that the (factored)-RBM likelihood is very peaked. It favors the detection of the peak.

Note that we assumed a perfect information game. However, it is possible to only observe one of the gazes at each time using the EXP3 algorithm from [1] or Bayesian optimization techniques [3].

3.3 Selection step

The aim of the selection is to obtain an "unweighted" approximate empirical distribution $\widehat{p}\left(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t},\mathbf{a}_{1:t}\right)$ of the weighted measure $\widetilde{p}\left(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t},\mathbf{a}_{1:t}\right)$. The basic idea is to discard the samples with small weights and multiply those with large weights. The introduction of this key step led to the first operational SMC method; see [7] for details of implementation of this black-box routine.

4 Experiments

In this section, three experiments are carried out to evaluate quantitatively and qualitatively the proposed approach. The first experiment provides comparisons to other control policies on a synthetic dataset. The second experiment, on a similar synthetic dataset, demonstrates how the approach can handle large variations in scale, occlusion and multiple targets. The final experiment is a demonstration on a real short hockey video. For the synthetic digit videos, we trained RBMs on the foveated images, while for the hockey video we trained factored-RBMs [23].

The first experiment uses 10 video sequences (one for each digit) composed using the MNIST dataset. Each sequence contains a moving digit, numbers in the background (to create occlusions) and salt and pepper noise (30%). The objective is to track and recognize the moving number. The gaze template was composed of $K = 4$ gazes, chosen so that gaze G1 was right at the center and gaze G4 nearby. The other gazes where placed in the periphery. The intention behind this choice of template was to evaluate whether hedge could learn where to look. The location of the template was initialized with optical flow.

We compare the policy learning algorithm against algorithms with deterministic and random policies. The deterministic policy chooses each gaze in sequence and in a particular pre-specified order, whereas the random policy selects a gaze uniformly at random. We adopted the Bhattacharyya distance in the specification of the observation model. A multi-class logistic regression model was trained to map the hidden units to the 10 digit classes. We used the transition prior as proposal for the particle filter.

Table 1 reports the comparison results. Tracking accuracy was measured in terms of the mean and variance (in brackets) over time of the distance between the target ground truth and the estimate; measured in pixels. The analysis highlights that the error of the learned policy is always below the error of the other policies. In most of the experiments, the tracker fails when an occlusion occurs for the deterministic and the random policies, while the learned policy is successful. This is very clear in the videos provided as additional material at: `http://www.youtube.com/user/anonymousML` **(We strongly recommend that readers look at these videos)**. The loss of track for the simple policies is mirrored by the high variance results in Table 1 (experiments 0, 1, 4, and so on). The average mean and variance errors (last column of Table 1) make clear that the proposed strategy for learning a gaze policy can be of enormous benefit.

Table 1: Tracking error on several video sequences using different policies for gaze selection.

	0	1	2	3	4	5	6	7	8	9	Avg.
Learned policy	**5.65** (0.58)	**2.82** (1.06)	**1.58** (1.22)	**2.31** (0.40)	**3.78** (0.77)	**1.92** (4.03)	**1.96** (0.48)	**4.82** (2.20)	**3.36** (13.50)	**1.80** (0.62)	**3.00** (2.48)
Deterministic policy	76.88 (1768.00)	34.83 (957.93)	1.92 (2.08)	3.60 (5.07)	55.90 (1988.21)	3.17 (10.83)	2.63 (2.48)	50.64 (4074.73)	52.54 (1364.43)	100.12 (4092.16)	38.22 (1426.59)
Random policy	51.90 (3936.27)	31.07 (1956.23)	1.76 (1.62)	3.59 (9.07)	88.78 (8516.38)	64.60 (5886.32)	79.28 (4036.75)	93.00 (4034.64)	93.63 (3874.49)	98.17 (6030.83)	60.58 (3828.26)

Figure 4 provides some anecdotal evidence for the policy learning algorithm, but we again encourage readers to watch the videos to better appreciate the behavior of the algorithms. The top sequence shows the position of the target (in green) and the particle filter estimate of its location (in red) over time. The middle sequence illustrates how the policy changes over time. In particular, it demonstrates that hedge can effectively learn where to look in order to improve tracking performance. This improvement in tracking also results in improvements in classification (see online videos). The classification results over time are shown in the third row.

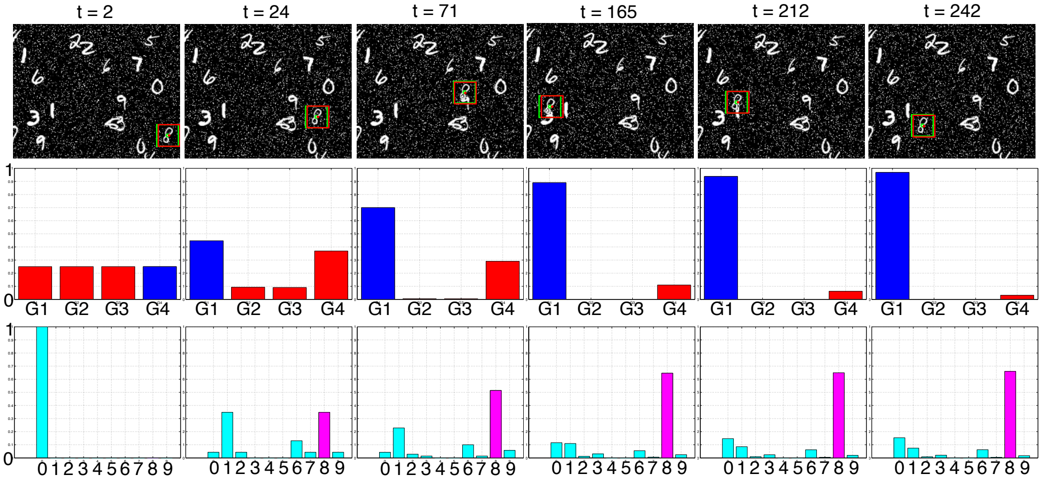

Figure 4: *Tracking and classification accuracy results with the learned policy. First row: position of the target in green and tracker estimate in red over time. Second row: policy distribution over the four gazes; hedge clearly converges to the most reasonable policy. Third row: cumulative class distribution for recognition.*

The second experiment addresses a similar video sequence, but tracking multiple targets. The image scale of each target changes over time, so the algorithm has to be invariant with respect to these scale transformations. In this case, we used a mixture proposal distribution consisting of motion detectors and the transition prior. We also tested a saliency proposal but found it to be less effective than the motion detectors for this dataset. Figure 5 (top) shows some of the video frames and tracks. The accompanying online video allows one to better appreciate the performance of the algorithm in the presence of occlusions. The final experiment was to track a hockey player in an actual video. The results are shown in Figure 5 (bottom) and the accompanying video.

5 Conclusions and future work

We have proposed a decision-theoretic probabilistic graphical model for joint classification, tracking and planning. This model is motivated by the dual-pathway (ventral and dorsal) architecture in

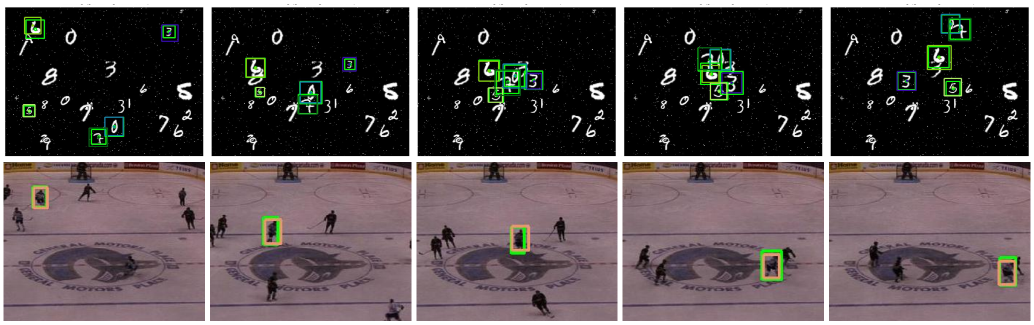

Figure 5: *(Top) Multi-target tracking with occlusions and changes in scale on a synthetic video. (Bottom) Tracking a player in a hockey video.*

neuroscience. The experiments demonstrate the significant potential of this approach. This can be better appreciated by watching the online videos.

There are many routes for further exploration. First, in this work we pre-trained the (factored)-RBMs. However, existing particle filtering and stochastic optimization algorithms could be used to train the RBMs online. Following the same methodology, we should also be able to adapt and improve the target templates and proposal distributions over time. The theoretical properties of these algorithms, in conjunction with policy learning, should be established.

Second, we used the activations of RBMs for a single gaze to train a second layer logistic classifier for digits, but this does not give the best recognition performance and does not exploit the correlation between gazes over time. However, in future work we plan to accumulate and integrate sequences of gazes to improve the classification performance and to emulate the learning of object representations in humans, as recently shown in [16].

Third, deployment to more complex video sequences will require more careful and thoughtful design of the proposal distributions, transition distributions, control algorithms, continuous template models, data-association and motion analysis modules. Fortunately, many of the solutions to these problems have already been engineered in the computer vision, tracking and online learning communities. Admittedly, much work remains to be done.

Saliency maps are ubiquitous in visual attention studies. Here, we simply used standard saliency tools and motion flow in the construction of the proposal distributions for particle filtering. There might be better ways to exploit the saliency maps, as neurophysiological experiments seem to suggest [10].

One of the most interesting avenues for future work is the construction of more abstract attentional strategies. In this work, we focused on attending to regions of the visual field, but clearly one could attend to subsets of receptive fields or objects in the deep appearance model.

Acknowledgments

We would like to thank Ben Marlin, Kenji Okuma, Marc'Aurelio Ranzato and Kevin Swersky. This work was supported by CIFAR's Neural Computation and Adaptive Perception program.

References

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. Technical Report NC2-TR-1998-025, NeuroCOLT2 Technical Report Series, 1998.

[2] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167 – 175, 2003.

[3] 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 TR-2009-023, University of British Columbia, Department of Computer Science, 2009.

[4] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, 2006.

[5] K. Chaudhuri, Y. Freund, and D. Hsu. A parameter-free hedging algorithm. In *Advances in Neural Information Processing Systems*, 2009.

[6] J. Colombo. The development of visual attention in infancy. *Annual Review of Psychology*, pages 337–367, 2001.

[7] A. Doucet, N. de Freitas, and N. Gordon. Introduction to sequential Monte Carlo methods. In A. Doucet, N. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[8] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[9] B. Girard and A. Berthoz. From brainstem to cortex: Computational models of saccade generation circuitry. *Progress in Neurobiology*, 77(4):215 – 251, 2005.

[10] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg. The representation of visual salience in monkey parietal cortex. *Nature*, 391:481–484, 1998.

[11] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[12] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[13] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254 –1259, 1998.

[14] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. Le-Cun. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition*, pages 1605–1612, 2009.

[15] U. Köster and A. Hyvärinen. A two-layer ICA-like model estimated by score matching. In *International Conference on Artificial Neural Networks*, pages 798–807, 2007.

[16] H. Larochelle and G. Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Advances in Neural Information Processing Systems*, 2010.

[17] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, 2009.

[18] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678, 2006.

[19] K. Okuma, A. Taleghani, N. de Freitas, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[20] B. A. Olshausen, C. H. Anderson, and D. C. V. Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13:4700–4719, 1993.

[21] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[22] E. O. Postma, H. J. van den Herik, and P. T. W. Hudson. SCAN: A scalable model of attentional selection. *Neural Networks*, 10(6):993 – 1015, 1997.

[23] M. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Computer Vision and Pattern Recognition*, pages 2551–2558, 2010.

[24] R. A. Rensink. The dynamic representation of scenes. *Visual Cognition*, pages 17–42, 2000.

[25] M. Rosa. Visual maps in the adult primate cerebral cortex: Some implications for brain development and evolution. *Brazilian Journal of Medical and Biological Research*, 35:1485 – 1498, 2002.

[26] K. Swersky, B. Chen, B. Marlin, and N. de Freitas. A tutorial on stochastic approximation algorithms for training restricted Boltzmann machines and deep belief nets. In *Information Theory and Applications Workshop*, pages 1–10, 2010.

[27] G. Taylor, L. Sigal, D. Fleet, and G. Hinton. Dynamical binary latent variable models for 3D human pose tracking. In *Computer Vision and Pattern Recognition*, pages 631–638, 2010.

[28] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine learning*, pages 1096–1103, 2008.

[29] M. Welling, M. Rosen-Zvi, and G. Hinton. Exponential family harmoniums with an application to information retrieval. *Advances in neural information processing systems*, 17:1481–1488, 2005.