CIRA 2009, Korea, December 15-18, 2009

Speeding up Top-Down Attention Control Learning by Using Full Observation Knowledge

N. Noori, M. Nili Ahmadabadi, M. S. Mirian and B.N. Araabi

***Abstract*—we present a general mathematical description of the top-down attention control problem. Three important components are identified in the model: context extraction, attention focus and decision making. The context gives a coarse blurry representation of the whole input; the attention module models the focus of attention on a limited part of input, and the decision making component accounts the final decision of the agent for its motory actions. In order to achieve a faster convergence of attention learning in the online phase, an offline optimization step is performed in advance. To do so, we incorporate the knowledge of a full observer agent that has approximately learned the optimal decision making of the task. The simulation results show that by employing our algorithm, the learning speed is improved.**

I. Introduction

Attention mechanisms act as dynamic filters that choose the most relevant parts of the information for detailed processing. The selection of useful information depends on the state of the agent in the environment; attention is not a static selection of input dimensions. Therefore, the attention problem can be best described as the question of *where* to look at the sensory input, and then according to *what* is perceived there, what to do next.

Attention is a demand in robotic applications where limited processing power is a bottleneck for the growing number of sensors. Without attention, it is impossible to process the entire visual information that is estimated to be on the order of $10^7 - 10^8$ bits per second. As a fundamental part in visual attention, gist or context of a scene is the amount of perceptual and semantic information that observers comprehend quickly within a glance (about 200 ms). Oliva and Torralba in [5] have proposed the *Spatial Envelope* model using only unlocalized amplitude information, i.e. global features, obtained from simple computations. In [8] global-context features and top-down influences are combined with bottom-up saliency in a Bayesian framework to predict likely image regions that are fixated by human observers in the visual search task. In this model, the context information is used to remedy the problem of huge search space by providing a prior on what to look for, and where to look for it [7]. The inspiration of this model is that in the real world the scene context modulates where eye movement should be directed.

On the other hand, based on Yarbus's experiments [15] attention control and learning are interleaved cognitive processes that are exploited for optimal decision making. The agent in a highly complex, dynamic and nondeterministic environment, must learn when and how to shift its focus of attention to meet its needs. The changes of attention focuses and motory decision makings can be coupled in a unified optimization problem. It seems that *Reinforcement Learning (RL)* methods are proper tools for learning top-down attention control through interaction with the environment. However, simultaneous learning of attention shifts and the task can be too slow and even it cannot converge without proper selection of input subsets as attention shifts.

In this work, we will describe attention control problem in a general mathematical form. We introduce context extraction in our model which resembles gist extraction in visual problems. We incorporate task knowledge in the form of the mind of a *learned full observer agent*, i.e. an agent that has learned a nearly optimal policy for the task. Using this knowledge we optimize the context extraction function in an offline stage before interactive learning of attention. We also design the possible attention shift actions using both the task knowledge and the designed context function. Primary simulations are conducted and the results justify the feasibility of our algorithm in speeding up attention learning.

II. The Attention Control Problem

A. Notations

In the rest of the paper, we will denote a vector by $\underline{V} = (v_1, v_2, \ldots, v_n) \in (V_1 \times V_2 \times \ldots \times V_n)$, its number of dimensions by $|\underline{V}| = n$, and the set of its basic vectors by the operator $D(\underline{V}) = \{V_1, V_2, \ldots, V_n\}$. We also denote sets by bold letters $\boldsymbol{v}$, number of elements in $\boldsymbol{v}$ by $|\boldsymbol{v}|$ and the space spanned by the subset $\boldsymbol{v} = \{V_{i_1}, V_{i_2}, \ldots, V_{i_m}\} \subseteq D(\underline{V})$ by $\underline{v} = (v_{i_1}, v_{i_2}, \ldots, v_{i_m}) = Proj(\underline{V}, \boldsymbol{v}) \in (V_{i_1} \times V_{i_2} \times \ldots V_{i_m})$.

B. The Problem

Assume that the input sensory information that is available to the agent has $|\underline{S}|$ dimensions, i.e. $\underline{S} = (s_1, s_2, \ldots, s_{|\underline{S}|})$ composing the space $\mathcal{R}^{|\underline{S}|}$. This is a massive amount of information that due to the limits of the agent's processing power cannot be processed in detail at the same time. The agent can either perceive *the whole input* ($\underline{S}$) and perform some *global*, *simple*, *light* computations like gist computations, or it can perceive part of input and perform detailed computations on them (*focus attention and processing resources onto small interesting regions*). In our modeling the former is called *context extraction* and the later is called *attention shifts*.

The authors are with the Robotics and AI Lab, Control and Intelligent Processing Center of Excellence, Department of Electrical and Computer Engineering, University of Tehran. Tehran, Iran. (email: {mnili, araabi, mmirian} @ut.ac.ir, noori.narges@gmail.com,Tel: +9821-8208-4950)

978-1-4244-4809-8/09/$25.00 ©2009 IEEE

The agent interacts with the environment through a set of n *motory actions*, shown by $\boldsymbol{a} = \{a_1, a_2, \dots, a_n\}$. The response that the agent receives from the environment for this action is a *reward* representing the goodness of the action. The ultimate goal of the agent is to maximize its expected reward considering its processing limits.

The agent has a *mental state* that represents its imagination of its real situation in the environment. Let us denote the agent's mental state at time i by MS_i. Before doing any motory action, the agent must first gain sufficient information about its real state in the environment through perceiving its sensory input. At the beginning, the agent has absolutely no knowledge concerning its real state in the environment, i.e. $MS_0 = null$. Its first glance at the sensory input is meant to obtain a hazy idea of its situation, i.e. the *context* of the real state of the agent. Let's name the function that extracts the context f_C. This function is a mapping from the sensory space to the context space:

$$f_C\colon S_{sensory\ input} \rightarrow S_{context}$$
$$\underline{context} = f_C(\underline{S}), \underline{context} = (c_1, c_2, \dots, c_{|\underline{context}|}) \quad (1)$$

Then the mental state of the agent changes according to the context, i.e. $MS_1 = \underline{context}$. In order to tune this mental state further, the agent looks at a subset of dimensions of $\underline{S}$; it performs an attention shift. Let's assume f_A be the function that selects the attended subset ($\boldsymbol{s}_i$) according to the agent's mental state:

$$f_A\colon S_{mental\ state} \rightarrow S_{Atteniton}$$
$$\boldsymbol{s}_i = f_A(MS_{i-1}), \qquad \boldsymbol{s}_i \subseteq D(\underline{S}) \quad (2)$$

where $S_{Atteniton}$ is the space of all possible subsets of dimensions of $\underline{S}$:

$$S_{Atteniton} = power\ set\ of\ D(\underline{S}) = 2^{D(\underline{S})} \quad (3)$$

After each attention shift, the mental state of the agent changes according to what it has *observed* in the attended subset and its current mental state:

$$MS_i = f_{MS}(MS_{i-1}, \underline{Observation}_i)$$
$$\underline{Observation}_i = f_O(\underline{s_i}), \ \underline{s_i} = Proj(\underline{S}, \boldsymbol{s}_i) \quad (4)$$

where f_{MS} is the function that represents how mental state changes based on sequential attention shifts, and f_O is the observation function.

The agent performs several attention shifts, namely μ, until it obtains sufficient information from the environment and it gets ready to decide which motory action it can do. The number of attention shifts, μ, depends on the real state of the agent in the environment and the agent can perform at most $max_{attention}$ attention shifts. Let's f_D be the function that makes motory decision according to the agent's mental state:

$$f_D\colon S_{mental\ state} \rightarrow \boldsymbol{a} = \{a_1, a_2, \dots, a_n\},$$
$$a_{optimal} = f_D(MS_\mu), \qquad 1 \leq \mu \leq max_{attention} \quad (5)$$

This modeling of f_C and f_A confirms the "coarse to fine" hypothesis [4], e.g. the sufficient information needed to decide whether a scene belongs to a natural or manmade category, can be obtained by a coarse view, and a finer categorization requires more detailed processing of specific features.

After the agent has performed the motory action, for the sake of simplicity we assume that its mental state will become $null$, i.e. $MS_0 = null$. Moreover, we assume that the agent's real state in the environment does not change during the attention shifts between subsequent motory action selections. A schematic view of the above description of attention control scenario is given in Fig. 1.

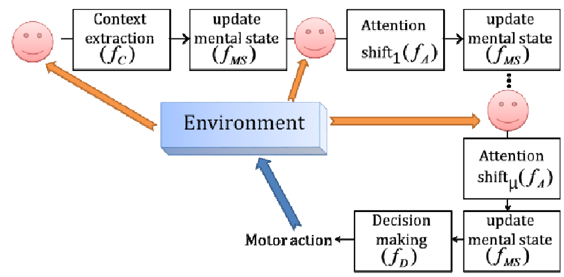

Fig. 1. The agent perceives information from environment by f_C and f_A.

To this point, we have introduced functions *context extraction* f_C, *attention shift* f_A, *decision making* f_D, and *mental state update* f_{MS}. Perceiving information from the environment is modeled by f_C and f_A. Therefore, the constraints that the agent's processing limits impose on receiving information from the environment are modeled by these two functions. Context extraction from sensory input requires simpler computations than processing the local focused attended input. As a result, the load of operations done on each dimension of the sensory input is lower in the context phase. Let's denote this load by $load_{context}$ and $load_{attention}$ in context extraction and attention shift phases respectively. Thus, we would have:

$$load_{context} \leq \ load_{attention} \quad (6)$$

and consequently:

$$\begin{cases} P_{l_C} = \dfrac{processing\ ability\ of\ the\ agent}{load_{context}} \\ P_{l_A} = \dfrac{processing\ ability\ of\ the\ agent}{load_{attention}} \end{cases} \rightarrow P_{l_A} \leq \ P_{l_C} \quad (7)$$

where P_{l_C} and P_{l_A} stand for the number of dimensions of the sensory input $\underline{S}$ that can be processed by the agent at context and attention phases. These constraints restrict f_C and f_A such that (1) will change into:

$$\underline{context} = f_C(\underline{s_C}), \qquad \underline{s_C} = Proj(\underline{S}, \boldsymbol{s}_C)$$
$$\boldsymbol{s}_C \subseteq D(\underline{S}), \qquad |\boldsymbol{s}_C| = P_{l_C} \quad (8)$$

and (3) will change into:

$$S_{Atteniton} = \{\boldsymbol{s} | \boldsymbol{s} \subseteq D(\underline{S}), |\boldsymbol{s}| = P_{l_A}\}$$
$$|S_{Atteniton}| = \binom{|\underline{S}|}{P_{l_A}} \quad (9)$$

III. The Learning Algorithm for the Attention Control Problem

To incorporate reinforcement learning method to learn attention control, we must define the agent's mental state f_{MS}, the set of its actions and rewards. Besides, we must identify how functions f_C, f_A and f_D are optimized in the learning method. We will first discuss the place of optimizing f_A and f_D in the learning and then we will return to f_C.

In the learning scheme, attention shifts and motory actions, f_A and f_D, must be learned simultaneously. The only thing that we can evaluate as critic is how proper is the motory actions chosen by the agent. Therefore, we should punish the agent for each attention shift and reward it for motory actions due to their properness related to the agent's state in the environment. Consequently, the set of actions will include both motory actions and possible selection of subsets of sensory input dimensions:

$$set\ of actions = \boldsymbol{a} \cup \{a(\boldsymbol{s_i})\ |\boldsymbol{s}_i \in S_{Atteniton}\} \quad (10)$$

where $a(\boldsymbol{s_i})$ stands for the attention shift action corresponded to the selection of the subset $\boldsymbol{s_i}$.

The agent's mental state function f_{MS} must be defined by the designer and we define it as:

$$\begin{aligned} MS_i &= f_{MS}\left(MS_{i-1}, \underline{Observation}_i\right) \\ &= \left(MS_{i-1}, a(\boldsymbol{s}_i), \underline{Observation}_i\right) \\ &= \left(\underline{context}, a(\boldsymbol{s_1}), \underline{obs}_1, a(\boldsymbol{s_2}), \underline{obs}_2, \ldots, \underline{obs}_{max_{attention}}\right) \end{aligned} \quad (11)$$

where if the agent performs $\mu < max_{attention}$ attention shifts, for $\mu \le i \le max_{attention}$ we will have $\underline{obs}_i = null$. Accordingly, the optimum μ in different states of the environment is also learned.

In the reinforcement learning algorithm, all of the possible states and actions must be explored which results in time and space complexity of $\Theta(\#\,states\ \times \#\,possible\ actions)$. As a result, when all members of $S_{Atteniton}$ are used as attention shift actions in (10), for large values of $|S_{Atteniton}| = \binom{|\underline{S}|}{P_{l_A}}$ referred to (9), learning would be too slow. Therefore, instead of using all members of $S_{Atteniton}$ as attention shift actions, we will design them and select some of $\binom{|\underline{S}|}{P_{l_A}}$ subsets as attention shifts through an offline optimization algorithm, that we call *Offline Optimum Subset Selection (OOSS)*. Let's denote the number of selected attention subsets by k. Thus, the set of actions in (10) contains $k + n$ actions.

Now we discuss optimization of the context extraction function f_C. Suppose that we wanted to learn it simultaneously with f_A and f_D. Recalling that the input argument of f_C is $\underline{s}_C = Proj\left(\underline{S}, \boldsymbol{s}_C\right)$ referred to (8), time and space complexity of learning would be in the order of:

$$\Theta(\text{attention control learning}) = \sum_{i=0}^{max_{attention}} q_S^{P_{l_C}} \times q_C^{|\underline{context}|} \times (q_O^{|\underline{obs}|} \times k)^i \times n \quad (12)$$

where q_S, q_C and q_O are quantization levels of sensory space, context space and observation space respectively. The components in (12) account for decision making based on context and i attention shifts. In all components, the term $\Theta(f_C) = q_S^{P_{l_C}} \times q_C^{|\underline{context}|}$ stands for learning of the function f_C. But context were considered to provide a coarse representation of the environment. It was meant to provide a prior to what look at next; to guide the subsequent attention shifts. In addition, the context must be extracted with little effort (low $load_{context}$). As a result, we usually have $P_{l_C} = |\underline{S}|$ and $\boldsymbol{s_C} = D(\underline{S})$ which leads to very high $\Theta(f_C)$. This will severely slow down attention control learning. Taking into consideration the above characteristics of context function, we will exclude f_C from online reinforcement learning optimization. Thus, the term $\Theta(f_C) = q_S^{P_{l_C}} \times q_C^{|\underline{context}|}$ in (12) will be reduced to $q_C^{|\underline{context}|}$. Since we have eliminated f_C, i.e. the mapping from $\underline{s}$ to $\underline{context}$, from online reinforcement learning we have to design f_C that satisfies the desired characteristics of context. To reduce the burden of designer, instead of precise definition of f_C, we allow the designer to just define the family of f_C, e.g. quadratic, cubic, or quartic polynomials, and optimize its parameters, e.g. the polynomial's coefficients, in an offline stage. We denote the set of f_C parameters that are free to be optimized, by $\boldsymbol{s}_{f_C}$. In addition, in the case that $P_{l_C} < |\underline{S}|$, we select $\boldsymbol{s_C}$ by our *OOSS* algorithm.

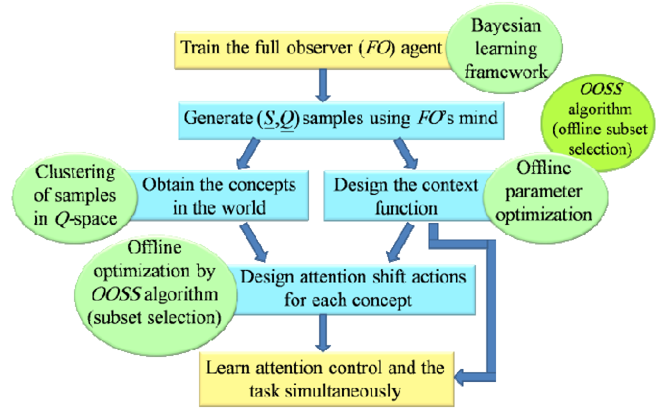

Fig. 2. The attention control learning algorithm; the online interactive and offline processing parts are shown in yellow and blue respectively. The methods used in each stage are in green circles.

We have discussed that before interactive (online) attention control learning, the attention subsets $\boldsymbol{s}_i$, the context parameters $\boldsymbol{s}_{f_C}$, and the context subset $\boldsymbol{s}_C$ must be optimized offline. To do this, let's look at an example of attention development in human driving. Novice drivers tend to look at the whole scene in front of them. They try to attend to everything around them. Therefore, they are very slow at making motory decisions and hence, they are trained in controlled environments. After a while, they gradually learn that they must attend to specific visual locations such that experienced drivers have specific eye fixation patterns [14]. We use this stage of slow, everything noticing period in human learning as an inspiration to incorporate the knowledge of a *full observer agent*. As its name suggests this agent has learned *to some extent* the optimal decision making of the task while it had sufficient resources to look at all of its sensory information $\underline{S}$.

The overall picture of our algorithm is shown in Fig. 2.

A. $(\underline{S}, \underline{Q})$ samples as the full observer agent's knowledge

We train the *full observer* agent with state space on $\underline{S}$ and the action space composed of motory actions $\boldsymbol{a}$. This is done by the continuous Bayesian based RL framework proposed in [3]. In this framework, the probability of visiting different states of the world ($\underline{S}$) is implicitly maintained in the mind of the *full observer* agent. Using this probability and the learned motory action values, we sample the $\underline{S}$ space and then compute their Q-values to obtain N_{sample} samples of the following form:

$$\underline{S}^i = (s_1, s_2, \dots, s_{|\underline{S}|})^i$$
$$\underline{Q}(\underline{S}^i, \boldsymbol{a}) = (Q(\underline{S}^i, a_1), Q(\underline{S}^i, a_2), \dots, Q(\underline{S}^i, a_n)) \quad (13)$$
$$sample^i = (\underline{S}^i, \underline{Q}(\underline{S}^i, \boldsymbol{a}))$$

i.e. each sample is vectors $\underline{S}^i$ and $\underline{Q}(\underline{S}^i, \boldsymbol{a})$ paired together.

B. Offline Optimum Subset Selection (OOSS) Algorithm

The input to this algorithm is a set of m paired vectors (samples) in the form ($1 \le i \le m$):

$$\left(\left(x_1, x_2, \dots, x_{|\underline{X}|}\right)^i, \left(y_1, y_2, \dots, y_{|\underline{Y}|}\right)^i\right) = (\underline{X}^i, \underline{Y}^i) \quad (14)$$
$$\underline{X}^i \in (X_1 \times X_2 \times \dots \times X_{|\underline{X}|})$$

The output of this algorithm is a subset $\boldsymbol{x} = \{X_{j_1}, X_{j_2}, \dots, X_{j_P}\} \subseteq D(\underline{X})$ of P best features of $\underline{X}$, i.e the most relevant dimensions of $\underline{X}$ related to $\underline{Y}$.

Selection of the best feature subset in an N ($N = |\underline{X}|$) dimensional space demands for examining $2^N - 1$ possibilities. Searching such a space is computationally exhaustive even for medium size feature spaces. In order to search the feature space more efficiently, the feature selection task is modeled as an expected reward maximization problem in an nth order Markov Decision Process (MDP). This MDP model drastically reduces the computational complexity of the problem. In this model, i.e. the nth order MDP, the current state $state_i$ is represented merely by n last selected features and the actions are selection of a feature. It is worth mentioning that the MDP model and the offline agent used here are just for this offline optimization algorithm and have no relationship with the discussion in other sections. The idea of this algorithm is from [2] and the only thing that we have revised in this algorithm is the measure for evaluation of selected features since we use the *mutual information* (*MI*) criterion.

The key property of *MI* is that if there is a nonlinear function between $\underline{x}$ and $\underline{Y}$ such that $\underline{Y} = f(\underline{x})$, the *MI* achieves its maximum value. On the other hand, if $\underline{x}$ and $\underline{Y}$ are independent *MI* becomes zero [13]. Furthermore, the importance of the set of $\underline{x}$'s features, i.e. $D(\underline{x})$, are jointly considered in the *MI* criterion. As we wanted to find the most relevant features of $\underline{X}$ related to $\underline{Y}$, *MI* is a good choice for superiority measure. To calculate the superiority measure (reward), first the $\underline{X}$ parts of all samples are projected into the space defined by the selected features, i.e. $\underline{x}_i = Proj(\underline{X}^i, \boldsymbol{h})$ with $1 \le i \le m$ and then the superiority measure (reward) is obtained by [13]:

$$r = MI((\underline{x}_i, \underline{Y}^i)_{i=1}^m) \quad (15)$$

C. Context Function Optimization ($\boldsymbol{s}_{f_C}$ and $\boldsymbol{s}_C$)

The context function f_C is defined by context parameters $\boldsymbol{s}_{f_C}$, and the context subset $\boldsymbol{s}_C$. We optimize $\boldsymbol{s}_C$ by *OOSS* algorithm with the whole N_{sample} samples $(\underline{S}^i, \underline{Q}(\underline{S}^i, \boldsymbol{a}))$ as the input. Recalling that the free parameters of f_C are in the set $\boldsymbol{s}_{f_C}$, we can use any optimization algorithm like genetic algorithm (*GA*) to find the optimum values of $\boldsymbol{s}_{f_C}$. But the important point is the superiority measure to assess different values of parameters in $\boldsymbol{s}_{f_C}$. Because of the discussed properties of *MI* criterion, we suggest using *MI* in order to evaluate the current values of parameters in $\boldsymbol{s}_{f_C}$ searched by the optimization algorithm (*GA* or *PSO*). Therefore, *MI* is calculated on $(\underline{context}^i, \underline{Q}(\underline{S}^i, \boldsymbol{a}))_{i=1}^{N_{sample}}$ where $\underline{context}^i$ is obtained by:

$$sample^i: \underline{context}^i = f_C(\underline{s_C}^i), \underline{s_C}^i = Proj(\underline{S}^i, \boldsymbol{s}_C) \quad (16)$$
$$1 \le i \le N_{sample}$$

and the free parameters of f_C ($\boldsymbol{s}_{f_C}$) are set to be equal to values found by the optimization algorithm (*GA* or *PSO*).

D. Selection of Attention subsets $\boldsymbol{s}_i$ in f_A

In order to select some attention subsets that looking at them is equivalent to performing attention shifts, there are two main questions that must be answered:

1- How many different subsets (k) are required?
2- How these subsets must be selected as possible attention shifts? (17)

To answer the above questions, let us introduce associative concepts in functionality space [1]. Consider action Q-value space $\underline{Q}(\underline{S}, \boldsymbol{a}) = (Q(\underline{S}, a_1), Q(\underline{S}, a_2), \dots, Q(\underline{S}, a_n))$ that we call the agent's functionality space. According to the definition of functional concepts in [9], similarity in action-value vectors maps them to the same concept. In other words, the concept $concept_i$ is composed of all states in perceptual space ($\underline{S}$) with almost equal action-value vectors, defined as follows:

$$concept_i = \{\underline{S} | \left\|\underline{Q}(\underline{S}, \boldsymbol{a}) - \underline{Q}(concept_i, \boldsymbol{a})\right\| \le \varepsilon, \varepsilon \ge 0\} \quad (18)$$

where $\underline{Q}(concept_i, \boldsymbol{a})$ is the functionality vector representing the $concept_i$. With this definition, concepts can be obtained by clustering samples in the functionality space, i.e. clustering $\underline{Q}$ part of samples, see Fig. 3.

Recall that attention control acts as *dynamic* selection of useful input information. Equivalently, the optimum $\boldsymbol{s}_i$ depends on the state of the agent in the environment. As an example, in an urban driving task, if we see a school caution sign in our first attention shift, the next region that we will attend would be the scene in front of us to take care of students that may pass across the street. But if in our first attention shift we see a turn right sign we would then attend to the rightmost region of the scene in front of us as our next attention shift. Therefore, using the idea of functional concepts, for two states that belong to different functional

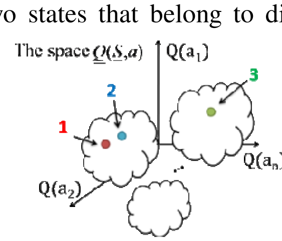

Fig. 3. Samples in the functionality space. Three functional concepts, corresponding to three clusters are shown. States $\underline{S}^1$ and $\underline{S}^2$ belong to the same concept because their Q vector are in the same cluster.

concepts, namely $\underline{S}^1$ and $\underline{S}^3$ in Fig. 3, the optimum attention shifts $\boldsymbol{s}_i^{\,1}$ and $\boldsymbol{s}_i^{\,3}$ that are done after context extraction, are different. Consequently we answer the questions in (17) by designing one attention subset for each functional concept, i.e. $k = number\ of\ concepts$. However, the design of more than one attention subset for each functional concept is a problem that we consider as our future work. Table I shows our algorithm for selecting the attention subsets $\boldsymbol{s}_i$.

TABLE I
SELECTING THE ATTENTION SUBSETS $\boldsymbol{s}_i$ IN f_A

1- *find the best clustering of* N_{sample} *samples* $\underline{Q}(\underline{S}^i, \boldsymbol{a})$ *namely* $cluster_1$, $cluster_2$,..., $cluster_m$

2- $k \leftarrow m$, *where k is the number of attention subsets* $\boldsymbol{s}_i$

3- *for each* $cluster_i$*(i'th functional concept)*

a. $\{sample^j\}^{cluster_i} \leftarrow$ ***samples in*** $\boldsymbol{cluster_i}$
$sample^j = (\underline{S}^j, \underline{Q}(\underline{S}^j, \boldsymbol{a}))$: $\underline{Q}(\underline{S}^j, \boldsymbol{a}) \in cluster_i$

b. *for samples in* $\{sample^j\}^{cluster_i}$ *compute context*
$\underline{context}^j = f_C(\underline{s_C}^j), \underline{s_C}^j = Proj(\underline{S}^j, \boldsymbol{s}_C)$

3-1- *select* P_{l_A} *dimensions as* $\boldsymbol{s}_i$ *for this concept by OOSS*

c. *use samples* $\{sample^j\}^{cluster_i}$ *as input to OOSS*

d. *use MI between vectors* $[\underline{context}^j, \underline{S}^j]^*$ *and* $\underline{Q}(\underline{S}^j, \boldsymbol{a})$ *as superiority measure in OOSS.*

* vectors $\underline{x}$ and $\underline{y}$ are concatenated to a single vector $\underline{z}$ by: $\underline{z} = [\underline{x}, \underline{y}]$

In line 1, the N_{sample} samples $(\underline{S}^i, \underline{Q}(\underline{S}^i, \boldsymbol{a}))$ are clustered in their $\underline{Q}(\underline{S}^i, \boldsymbol{a})$ space, and each cluster is considered as a functional concept. However, the number of clusters is unknown and it depends on the clustering algorithm that is used. Suppose that m clusters are formed. Since we have decided to design one attention subset for each functional concept we would use $k = m$, and k attention subsets $\boldsymbol{s}_i$ will be selected from the whole $\binom{|\underline{S}|}{P_{l_A}}$ subsets as follows. In line 3 through 3-1-d, for each concept (cluster) an attention subset $\boldsymbol{s}_i$ is selected using *OOSS* algorithm. In lines 3-a and 3-b the required inputs for *OOSS* are prepared. Samples in the *i*'th cluster are gathered in $\{sample^j\}^{cluster_i}$ in line 3-a. This way the selection of $\boldsymbol{s}_i$ for *i*'th concept would be based on samples in the *i*'th cluster. To bring into play the information obtained from context extraction, the $\underline{context}$ is computed for samples in line 3-b, recalling that the context function parameters were previously optimized. In line 3-1, the optimization algorithm *OOSS* is called. The paired vectors $(\underline{S}^j, \underline{Q}(\underline{S}^j, \boldsymbol{a}))$ in $\{sample^j\}^{cluster_i}$ are used as input to *OOSS*, in line 3-1-c. The superiority measure in *OOSS* algorithm, i.e. reward signal in (15), is computed as the *MI* between vectors $[\underline{context}^j, \underline{S}^j]$ and $\underline{Q}(\underline{S}^j, \boldsymbol{a})$ for samples in $\{sample^j\}^{cluster_i}$.

IV. EXPERIMENTS AND RESULTS

To verify our algorithm we used the same dataset that is used in [12]. This dataset includes movies from campus sites in Ahmanson Center for Biological Research (ACB). These video clips have been divided into nine segments. Images within each segment look similar to a human observer. Sample images from each site are shown in Fig. 4. The sensory input, the set of motory actions and the environment rewarding are as follows. The images are divided into $row \times column$ regions with $row = 2$ and $column = 9$. Then for each region the percentage of *red*, *green*, and *blue* colors are computed. The resulted $row \times column \times 3 = 54$ dimensional vectors are considered as the sensory inputs $\underline{S}$. We use 200 samples for each segment. For samples in each segment some motory actions are considered to be good actions by the environment, indicated by one in Fig. 4. For each sample some of these good actions are best. For example, suppose that two images img_1 and img_2 belong to the first segment. Therefore, actions a_1 and a_2 are good actions for both of them. The environment chooses randomly one of the good actions a_1 and a_2 as the best action for img_1, namely a_1, and assigns it to img_1. Let's a_2 be the best action assigned to img_2. Then a_1 and a_2 are always best actions for img_1 and img_2 respectively. The reward that the agent receives for different motory actions is as follow:

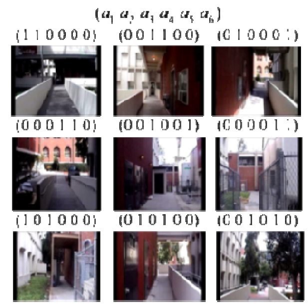

Fig. 4. Examples of images in each of 9 segments of ACB (from [12] downloaded from [16]). Good actions for samples in each concept are indicated by one.

$$Reward\,(\underline{S}, a_i) = \begin{cases} max_{reward} & if\ a_i\ is\ a\ good\ and\ also\ the\ best\ action\ for\ \underline{S} \\ max_{reward} \times 0.6 & if\ a_i\ is\ a\ good\ action\ in\ (\underline{S}),\ but\ it's\ not\ the\ best \\ punishment & if\ a_i\ is\ not\ rewarding\ in\ concept(\underline{S}) \end{cases}$$

where $max_{reward} = -punishment = 20$. The attention control agent is punished by -7 for each attention shift, and there are 6 motory actions, i.e. $n = 6$.

The agent's processing limits are set to be $P_{l_C} = P_{l_A} = 9$. The context function f_C is the decision made by a local expert trained over the partial sensory input $\underline{s_C} = Proj(\underline{S}, \boldsymbol{s}_C)$ as in [11]. However, instead of training local experts on partial sensory input, we use the knowledge of the *full observer* agent to estimate their decisions [10]:

$$Q'_m = Q(\underline{s_i}, a) = \iint_{m \in s} P(s|m) \times Q(\underline{S}, a) \qquad (19)$$

and the set of free parameters of f_C, $\boldsymbol{s}_{f_C}$, would be empty.

We use two different learning spaces: decision space and perceptual (sensory) space. The motory actions' *Q*-values of local experts on $\underline{s_i}$ is employed as the decision space, $\underline{Observation}_i = f_O(\underline{s_i}) = \underline{Q}(\underline{s_i}, \boldsymbol{a})$, which is estimated based on (19), and the identity function $\underline{Observation}_i = f_O(\underline{s_i}) = \underline{s_i}$ will define the perceptual space.

Fig. 5-a compares attention control learning in perceptual and decision spaces respectively after the design of different parts by our algorithm. Since the results show the advantage of learning on decision space over perceptual space, we will use decision space in our following simulations.

To find the best clustering of samples we use silhouette criterion for different number of clusters and the clustering with maximum mean silhouette is chosen. The silhouette value for each sample is a measure of how similar that sample is to samples in its own cluster compared to samples in other clusters, and ranges from -1 to +1. As shown in Fig. 5-b, the number of clusters is chosen to be 9 ($k = 9$). In fact, the environment rewarding for 6 motory actions defines 9 functional concepts. Finally, we use $max_{attention} = 9$ as the maximum number of attention shifts.

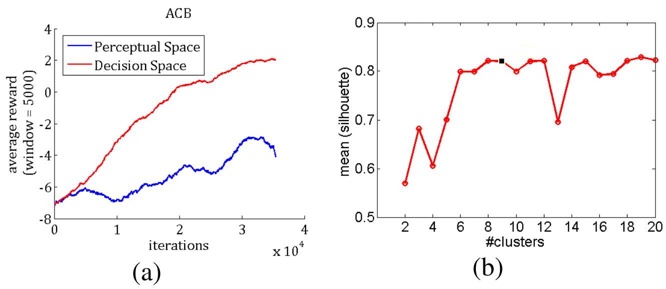

Fig. 5. a) Attention control learning in perceptual space ($f_O(\underline{s}_i) = \underline{s}_i$) and decision space $f_O(\underline{s}_i) = Q'_m$. b) The average silhouette values for different number of clusters. We select the clustering with 9 clusters.

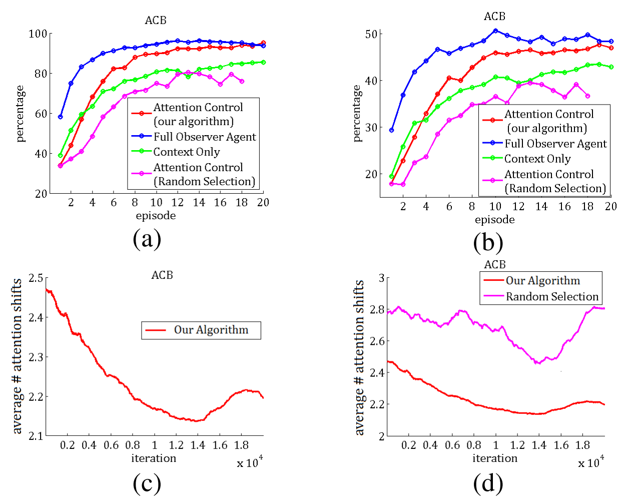

Fig. 6. percentage of samples that the action suggested by the agent is a) among one of the good actions for the sample. b) exactly the best action for the sample. Average number of attention shifts (window = 5000) c) using the subsets designed by our algorithm. d) comparison between random selection of subsets and subset selection by our method.

Fig. 6 depicts the results of our simulations. In Fig. 6-a and 7-b we have compared learning in four agents: attention control learning using subsets designed by our algorithm, the full observer agent, the agent using just the designed context subset $\boldsymbol{s}_C$, and attention control learning using subsets that are randomly chosen. The attention learning by our designed subsets finally reaches the performance of the full observer agent and it is better than other two agents, i.e. context only and attention on randomly selected subsets. This shows that careful design of attention shift actions speeds up learning. The average number of attention shifts for the agent designed by our method is shown in Fig. 6-c that reaches 2.2 out of $max_{attention} = 9$. It decreases over time that shows learning of useful information. This value is compared with that in the agent with randomly selected subsets in Fig. 6-d. The number of attention shifts doesn't decrease in this case which shows the advantage of our algorithm.

V. Conclusion and Future Work

In this paper we have presented a framework for attention control learning. We have discussed that learning must be done just for attention and decision making parts arguing that learning context extraction function is exhaustive. We also showed that we can reach faster learning by incorporating the knowledge of a full observer agent. This knowledge is used to design context function and after that the attention shift actions. Furthermore, we used the idea of functional concepts to guide our design of attention shifts.

To validate our method, we artificially built an environment that defines 9 functional concepts by 6 motory actions. The results are promising since the performance of the designed attention learner is comparable with that of the full observer agent and it's also better than random selection of attention subsets.

References

[1] H. Mobahi, M. Nili Ahmadabadi, B. N. Araabi, "Concept oriented imitation towards verbal human-robot interaction," *ICRA*, 2005.

[2] M. T. Harandi, M. Nili Ahmadabadi and B. N. Araabi, "Optimal local basis: a reinforcement learning approach for face recognition," *Journal of Computer Vision,* Springer, 2008.

[3] H. Firouzi, M. Nili, B. Araabi, "A Probabilistic Reinforcement-Based Approach to Conceptualization," *IJIT*, Vol. 3, pp 48-55, 2008.

[4] O. R. Joubert, G. A. Rousselet, D. Fize and M. Fabre-Thorpe, "Processing scene context: Fast categorization and object interference," *Journal of Vision Research,* Elsevior, 2007.

[5] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, pp. 145–175, 2001.

[6] A. Oliva, A. Torralba, "Building the gist of a scene: the role of global image features in recognition," *Brain Research*, vol. 155, 2006.

[7] P. Murphy, A. Torralba and W. T. Freeman, "Using the forest to see the trees: a graphical model relating features, objects and scenes," *NIPS*, Vancouver, BC, MIT Press, 2003.

[8] A. Torralba, A. Oliva, M. Castelhano, J. M. Henderson, "Contextual Guidance of Attention in Natural scenes: The role of Global features on object search," *Psychological Review*, vol. 113(4) 766-786, 2006.

[9] H.Vosughpour, M. Nili, Maryam S. Mirian, "Hierarchical functional concept formation using reinforcement learning," *Proceeding of WCECS, ICMLDA*, 2007.

[10] T. Jaakkola , S. P. Singh , M. I. Jordan, "Reinforcement learning algorithm for partially observable Markov decision problems," in *Advances in Neural Information Processing Systems*, 1995.

[11] M. S. Mirian**,** M. Nili Ahmadabadi, B. N. Araabi, R. R. Siegwart, "Comparing learning attention control in perceptual and decision space," LNAI 5395, pp. 242 – 256, 2009, Springer-Verlag.

[12] C. Siagian, L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *PAMI,* 2007.

[13] J.C. Principe, D. Xu, J. Fisher, "Information theoretic Learning," in *S. Haykin (Ed.), Unsupervised Adaptive Filtering*, Wiley, New York, 2000, pp. 265–319.

[14] G. Underwood, P. Chapman, N. Brocklehurst, J. Underwood and D. Crundall, "Visual attention while driving: sequences of eye fixations made by experienced and novice drivers," *Ergonomics*, 2003.

[15] A. L. Yarbus, "Eye movements during perception of complex objects," in *Eye Movements and Vision*, ed. L. A. Riggs, Plenum Press, New York, ch. 7, pp. 171–196. 1967.

[16] http://ilab.usc.edu/siagian/Research/Gist/Gist.html.