Institute for Research in Fundamental Sciences (IPM)

School of Cognitive Sciences (SCS),

Ph.D. Thesis

Interactive Learning of Task-Driven Visual Attention Control

By: Ali Borji

Advisors:
Dr. Babak Nadjar Araabi
Dr. Majid Nili Ahmadabadi

Spring 2009

Declaration

This dissertation was conducted under the supervision of Dr. Babak N. Araabi and Dr. Majid N. Ahmadabadi. The work submitted in this thesis is the result of original research carried out by myself, except where acknowledged. It has not been submitted for any other degree or award.

Ali Borji
May. 2009

Acknowledgement

I want to thank all the people who contributed to this thesis in one way or another. I wish to express profound gratitude to my supervisors, Dr. Babak N. Araabi and Dr. Majid N. Ahmadabadi. I thank Dr. Araabi for the opportunity and inspiration to conduct this research. I thank Dr. Ahmadabadi for his persistent guidance and enthusiasm. I wish to thank both Dr. Ahamadabadi and Dr. Araabi for the past, present and future opportunities that would not exist without their generosity and support. I also appreciate Prof. Hossein Esteky for his wise advices during the course of this study. I am grateful to Mr. Abdolhossien Abbassian for our fruitful discussions and the insights he gave me during my stay at IPM.

I also wish to thank all of my friends and colleagues at the robotic lab of university of Tehran for making the environment rewarding. In particular, I thank Mrs. Mirian for all the frivolous, earnest and always entertaining discussions that on several occasions elicited side projects.

Finally, special thanks go to my family who unconditionally supported me during all stages of my studies and without their love it was impossible to complete this work.

Ali Borji
May. 2009

Abstract

Humans are remarkably efficient to act reasonably in an environment based on perceptual information they receive. Vision is the most important sensor humans rely on, and that's the main reason of being the most-studied mode of machine perception in artificial intelligence (AI). Despite the huge active research in computer vision and robotics, many real-world visiomotor tasks which are easily performed by humans are still unsolved. Of special interest is designing efficient learning algorithms, in terms of high accuracy and low computational complexity, for enabling autonomous mobile robots to act in visual interactive environments where agents have to deal with natural outdoor scenes. This is actually much harder to tackle compared with controlled visual environments (e.g. indoor scenes) used in robotic laboratories. Example applications of visual learning are vision guided navigation, vision-based place recognition, grasping and object manipulation.

Recent trend in robotics is toward developing robots able to act autonomously in unknown, stochastic and partially observable environments. This desired quality makes interactive and online learning of visual representations and control policies an essential element. Such dynamic methods results in flexible solutions with less complexity and lower computational power. For a robotic agent to be able to act in a visual environment, it should map its visual perceptual space into physical actions also called visiomotor coordination, vision for action or purposive vision. In contrast with machine vision solutions which assume predefined and rich visual representations in mind of an agent, necessary representations could be learned while the agent performs physical actions for achieving a goal.

In this dissertation, I introduce several solutions for learning top-down and task-driven visual attention control in interactive environments and when natural scenes have to be dealt with. Agent should learn visual representations in concert with physical actions in order to do a complex task. This work is motivated by the paradigm of purposive vision and visual attention. The original contributions consist in the design of reinforcement learning algorithms that are applicable to visual spaces. Inspired by the spatial attention paradigm, our main aim is to design solutions for feature extraction at only subsets of image regions. Emphasis is on learning spatial attention along with motor actions.

Keywords: purposive vision, vision for action, top-down attention, visual attention, bottom-up attention, learning attention control, task-driven attention, reinforcement learning, RLVC, U-TREE, computer vision, cognitive robotics

Contents

List of Tables

List of Figures

Chapter 1
Introduction

Human vision can basically be divided into two main phases, low-level vision and high-level vision. Although the border between these two vision phases is not clearly defined, their main roles have already been established. Low-level vision starts with gathering visual information by means of the retinas in the order of 10^8 bits/second. The gathered information is then transmitted to the visual cortex where information about motion, depth, color, orientation, and shape is extracted. The high-level vision performs then its task on these extracted features. It is mainly responsible for recognizing scene contents by matching the representative scene features to a huge database of learned and memorized objects (i.e. object recognition).

Despite the huge amount of visual information to be processed and despite the combinatorial nature of the recognition task, it has been estimated that humans can recognize a large number of objects in less than 200 ms. The computational resources of the human brain can not entirely explain this surprising performance, since about 10^{11} neurons of our brain cannot process such amount of information in such a short time, given their slow response rate. This high performance speaks rather for the high efficiency of our vision system. The existence of a mechanism that selects only a reduced set of the available visual information for high level processing seems to be the most coherent explanation of the high performance of the human visual system. This mechanism is referred to as visual attention. Visual Attention helps people to extract the relevant information at an early processing stage. The extracted information is then directed to higher brain areas where complex processes such as object recognition take place. Restricting processes to a limited subset of the sensory data enables efficient processing.

1.1 Task Relevancy of Visual Attention

What is the relevant information to be attended? There is no general answer to this question because relevancy depends on situation. Without any goal in mind like in free-viewing tasks, certain cues like strong contrasts attract our attention. The saliency also depends on the spatial context in which objects are presented. In addition to the bottom-up cues, attention is also influenced by top-down cues, that means cues from higher brain areas like knowledge, motivations, internal states and emotions. These bottom-up and top-down attentional mechanisms interact together for controlling attention.

In addition to many psychophysical, and neurophysiological studies to unveil the mysteries behind attention, many computational models have also been proposed in the literature to inspire from this natural behavior for solving complex problems with the same characteristics like information overload. In computer vision, the computational complexity of numerous tasks like perceptual grouping and object recognition, which are known to be NP-Complete, represents a

fundamental obstacle for real-world applications. Thus, the visual attention paradigm is a highly relevant topic if we want to resolve the complexity issue in computer vision and robotics. Indeed, visual attention can be thought as a preprocessing step which permits a rapid selection of a subset of the sensory information. Once detected, the salient parts become the specific scene locations on which higher level computer vision tasks can focus.

1.2 Reinforcement Learning and State-Space Discretization

Reinforcement learning is a biologically-inspired computational framework that can generate nearly optimal control policies in an automatic way, by interacting with the environment. RL is founded on the analysis of a so-called reinforcement signal. Whenever the agent takes a decision, it receives as feedback a real number that evaluates the relevance of this decision. From a biological perspective, when this signal becomes positive, the agent experiences pleasure, and we can talk about a reward. Conversely, a negative reinforcement implies a sensation of pain, which corresponds to a punishment.

Now, RL algorithms are able to map every possible perception to an action that maximizes the reinforcement signal over time. In this framework, the agent is never told what the optimal action is when facing a given percept, nor whether one of its decisions was optimal. Rather, the agent has to discover by itself what the most promising actions are by constituting a representative database of interactions, and by understanding the influence of its decisions on future reinforcements.

Like many existing learning methods, RL suffers from the curse of dimensionality, requiring a large number of learning trials as state-space grows. But it has the ability to handle dynamic and non-deterministic environments. It is believed that curse of dimensionality can be lessened to a great extent by implementation of state abstraction methods and hierarchical structures. Moreover, incremental improvement of agent's performance becomes much simpler due to less number of states.

Several approaches for interactive discretization of state space have been proposed. Techniques using a non-uniform discretization are referred to as variable resolution techniques (Munos & Moore, 2002). The parti-game algorithm (Moore & Atkeson, 1995) is an algorithm for automatically generating a variable resolution discretization of a continuous, deterministic domain based on observed data. This algorithm uses a greedy local controller and moves within a state or between adjacent states in the discretization. When the greedy controller fails, the resolution of the discretization is increased in that state. The G algorithm (Chapman & Kaelbling, 1991), and McCallum's U-Tree algorithm (1996), are similar algorithms that automatically generate a variable resolution discretization by re-discretizing propositional techniques. Like parti-game, they both start with the world as a single state and recursively split it when necessary. The continuous U-Tree algorithm described in (Uther & Velso, 1998), extends these algorithms to work with continuous state spaces.

1.3 The Link between Task-driven Attention and State-Space Discretization

To perform a task, agents should be able to percept the environment and perform appropriate physical actions. Perceptual actions are available in several forms like where and what to look in visual modality. The main concern in learning attention is how to select the relevant information, since relevancy depends on the tasks and goals. In this thesis, we consider task relevancy of

visual attention and aim to extract spatial locations which help the agent to achieve its goals faster.

It is important that a solution for learning task-based visual attention control to take into account other relevant and interleaved cognitive processes like learning, decision making, action selection, etc. There are several biological evidences for this. It has been previously shown that attention and eye movements are context-dependent and task-specific (Yarbus, 1967). Previous experiences also influence attentional behaviors which indicate that attention control mechanisms can be learned (Maljkovic & Nakayama, 1994). Some neuropsychological evidences suggest that human beings learn to extract useful information from visual scenes in an interactive fashion without the aid of any external supervisor (Gibson & Spelke, 1983; Tarr & Cheng, 2003). Instead of attempting to segment, identify, represent and maintain detailed memory of all objects in a scene, there are evidences that claim our brain may adopt a need-based approach (Triesch et al., 2003), where only desired objects are quickly detected, identified and represented. Considering above evidences, in this work, we introduce a model to consider the influences of task, action, learning and decision making to control top-down visual attention.

From another perspective learning top-down attention is highly coupled with learning representations. Therefore the best way to derive visual attention mechanisms is to learn them in concert with visual representations. This is tightly relevant to an area of research known as state space discretization in reinforcement learning. Through this thesis we elaborate more on this relevancy.

Following from the above discussion, a breakthrough in modern artificial intelligence would be to design an artificial system that would acquire object or scene recognition skills based only on its experience with the surrounding environment. To state it in more general terms, an important research direction would be to design a robotic agent that could autonomously acquire visual skills from its interactions with an uncommitted environment in order to achieve some set of goals. Learning new visual skills in a dynamic, task-driven fashion so as to complete an a priori unknown visual task is known as the purposive vision paradigm. Our solutions in thesis are in the context of the purposive vision and extend this track of research by integrating concepts of purposive vision with visual attention. In the context of purposive vision, visual attention enhances them by reducing the computational complexity. In the context of visual attention, approaches for purposive vision propose valid and efficient solutions for learning task-driven visual attention control.

1.4 Objectives

Because of high importance and criticality of attention in computer vision and cognitive robotics applications, our aim in this project is to extend state-of-the art approaches for modeling attention by incorporating a learning mechanism. In another words, we are going to take the essence of this phenomenon in biology and incorporate it for artificial agents and robots. Although the problem is the same, because artificial agents are much different than humans in terms of their body, brain and the environment, the same solution might not be appropriate and therefore appropriate solutions has to be devised.

An agent dealing a complex task faces a huge amount of data from which a lot is irrelevant to its task. Therefore agent needs a mechanism or filter to bypass irrelevant information and pick up important and relevant information to perform its task. In many cases attention leads to more

efficient solutions but in some cases attention is critical meaning that without attention a task is not possible to be solved. An example of such a complex task is driving. When driving, our eyes capture a lot of information from the surroundings, knowing how and when we collect a piece of that information could be very useful in designing artificial driving agents. For naïve drivers, it is hard to attend and perform motor actions simultaneously at first but after training they learn to coordinate their actions and attentions and learn what to do and what to attend.

Our motivation for learning spatial attention is human eye movement behavior. Humans actively move their eyes over a scene for gathering relevant information for example for indentifying a face. This kind of eye movement is known as saccadic eye movements. We would like to generate such a scanpath of eye movements for an artificial agent. But here the relevancy is determined by the task demands. Based on studies in neuroscience of visual attention we propose a framework for learning top-down and task-driven visual attention control. We also bring rationales for the link of solutions with biology.

1.5 Contributions

Our contributions are shortly explained here and each one will be explained in detail in its own section in the thesis.

Associative Learning of Attention and Action Similar to humans and primates, artificial creatures like robots are limited in terms of allocation of their resources to huge sensory and perceptual information. Thus attention is regarded as the same solution as humans in this domain. Some sorts of our attentional behaviors are associative where we know what to attend in which situation. Motivated by this we formulate attention as an optimization problem in which the agent has to gain maximum reward while satisfying a constraint which is its information processing bottleneck. Reinforcement learning is then used to solve that optimization problem. A driving environment is simulated in that agent has to learn how to drive safely by attending to the right spatial locations and performing appropriate motor actions.

Biasing the bottom-up saliency-based model of visual attention A biologically-inspired model of visual attention known as basic saliency model is biased for object detection. It is possible to make this model faster by inhibiting computation of features or scales which are less important for detection of an object. To this end, we revise this model by implementing a new scale-wise surround inhibition. Each feature channel and scale is associated with a weight and a processing cost. Then a global optimization algorithm is used to find a weight vector with maximum detection rate and minimum processing cost. This allows achieving maximum object detection rate for real time tasks when maximum processing time is limited. A heuristic is also proposed for learning top-down spatial attention control to further limit the saliency computation.

Interactive Learning of Object-based Attention Control In this contribution, we propose a biologically-motivated computational model for learning task-driven and object-based visual attention control in interactive environments. In order to do a task an artificial agent should be able to perform motor and perceptual actions at the same time. Our model consists of three layers. First, in the early visual processing layer, basic layout and gist of a scene are extracted. The most salient location of the scene is simultaneously derived using the biased saliency-based bottom-up

model of visual attention. Then a cognitive component in the higher visual processing layer performs an application specific operation such as object recognition and scene understanding at the focus of attention. From this information, a state is derived in the decision making layer. Top-down attention in our model is learned by the U-TREE algorithm which successively grows a tree whenever perceptual aliasing occurs. Internal nodes in this tree check the existence of a specific object in the scene and its leaves point to states in the Q-table. Motor actions are associated with leaves. After performing a motor action, the agent receives a reinforcement signal from the critic. This signal is alternately used for modifying the tree or updating the action selection policy. A long-term memory component holds the bias signals of important task-relevant objects of the environment. Basic saliency-based model of visual attention is devised to consider processing costs of feature channels and image resolutions. To recognize objects, a recent and successful object recognition method, inspired by the hierarchical organization of the visual ventral stream, is used. The proposed model is evaluated on visual navigation tasks, where obtained results lend support to the applicability and usefulness of developed method for robotics.

Saliency maps for Attentive and Action-based Scene Classification This contribution proposes an approach for scene classification by extracting and matching visual features only at the focuses of visual attention instead of the entire scene. Analysis over a database of natural scenes demonstrates that regions proposed by the saliency-based model of visual attention are robust with image transformations. Classification results show that classification rate is nearly the same as when features are extracted over the entire scene but with feature extraction at a small fraction of scene regions. An approach for simultaneous learning of physical actions and representations known as RLVC is also extended by limiting it's SIFT search and extraction to only salient regions. Results prove that RLVC has still the same performance with a huge decrease in computational complexity. Overall, our results prove that efficient scene classification, in terms of reducing the complexity of feature extraction is possible without a significant drop in performance.

Interactive Learning of Space-based Visual Attention Control and Physical Actions A new method for learning top-down and task-driven visual attention control along with physical actions in interactive environments is proposed. Our method is based on the RLVC algorithm and adapts it for learning spatial visual selection in order to reduce computational complexity. Proposed algorithm also addresses aliasings due to not knowing previous actions and perceptions. Continuing learning shows our method is robust to perturbations in perceptual information. Proposed method also allows object recognition when class labels are used instead of physical actions. We have tried to gain maximum generalization while performing local processing. Experiments over visual navigation and object recognition tasks show that our method is more efficient in terms of computational complexity and is biologically more plausible.

1.6 Outline of the Thesis

Outline of this thesis is as follows. Chapter one, introduces the scope of the thesis and highlights its objective and the problems it is aimed to solve. Part one brings the background material and forms the basis for next chapters. Background literature is reviewed from two perspectives: visual attention and reinforcement learning and state-space discretization. Basics of visual attention

which involves behavioral and modeling studies are covered in chapters 2.1 and 2.2 respectively. Applications of attentional models in computer vision and robotics are also reviewed in chapter 2.3. Theoretical framework which is the basis of our solutions in this thesis are shown in chapter 3. Next sections show how our approaches are built upon these solutions.

Second part contains chapters 4 to 8. Chapter four considers learning associations between states on the environment and physical actions. This solution solves tasks where there is not a specific goal to be achieved. For example in a simulated driving environment, a driver agent has to learn to drive safely without collision. Chapter five is not based on RL and despite other methods does not target interactive learning. It forms a basis for chapter 6 for interactive learning of object based attention control. The idea behind the object based attention control is that an agent should find its state which is a natural scene and then map it to an appropriate action. The way to classify a scene here is to identify objects present in that scene. However, since segmentation of all the objects of a scene is costly we only limit the processing to few objects. The objects that have to be attended are learned in an interactive fashion. Since in this top-down approach, bottom-up biases are applied to a scene to find whether and object is present in a scene or not, these bottom-up biases are learned in a data-driven approach from a set of training images in the contribution of chapter 5.

Chapters 7 and 8 are devoted for interactive learning space-based attention and physical actions and are direct extensions to the RLVC. In chapter 7, computations of RLVC are limited to only salient regions which are to some degree stable over image transformations. Chapter 8, is a saccade learning approach known as S-Tree and only inspects a few spatial regions of the image. Figure 1.1 shows the dependency of chapters.

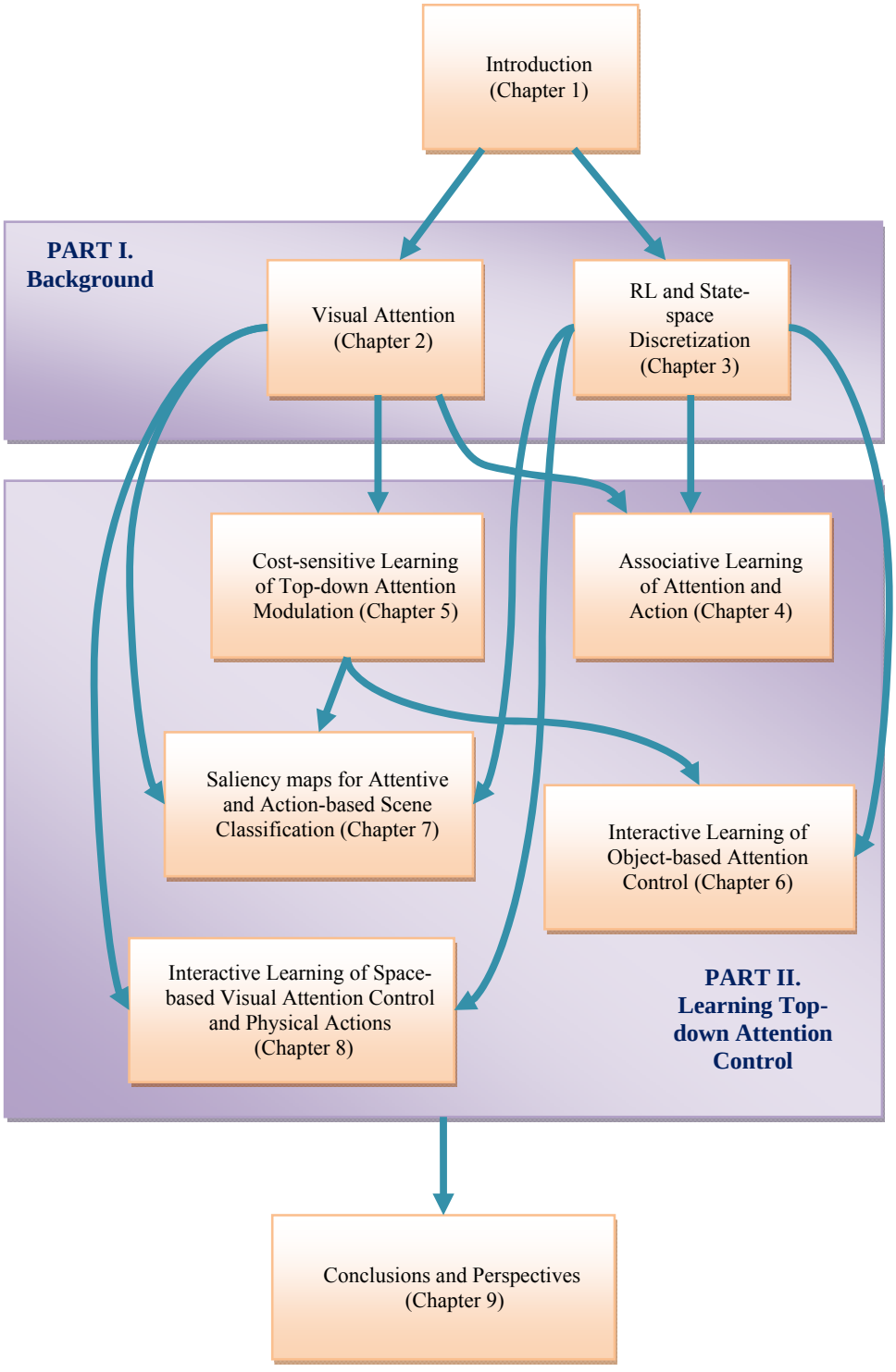


Figure 1.1 Outline of this thesis. Arrows represent the dependencies between the various concepts that are introduced in the chapters.

Part I
Background Material

Chapter 2
Visual Attention
2.1 Basic Concepts

2.1.1 Introduction

Information overload is one key problem in perception. We continuously acquire new information about our environment through our sensors. Processing this much information can be very expensive. Accordingly, brain can not process all of this data. Therefore, it makes decisions on which information is relevant and will be selected for further information processing and which will be filtered out. This process of selection and priorization of incoming information is called SELECTIVE ATTENTION.

Only a limited aspect of visual information is being processed by higher level of the visual system, demonstrating the limited capacity of the visual system, an experiment called *change blindness* has been introduced by (O'Regan, Rensink, and Clark, 1997). In this experiment subjects presented a visual scene as an image fail to detect significant changes in the image (e.g. a car disappears). Hence, VISUAL ATTENTION can be seen as the process for reducing the amount of incoming visual information to a smaller quantity which the brain can handle. As Itti (2000) argues, O'Regan and collegues experiment reveals something about the mechanisms used by humans to control where they deploy spatial or focal attention while they examine a scene.

Therefore, it can be said that visual attention controls the levels of information which will be selected and ensures that the selected information is relevant to further behavioral decisions.

2.1.2 Components of Visual Attention

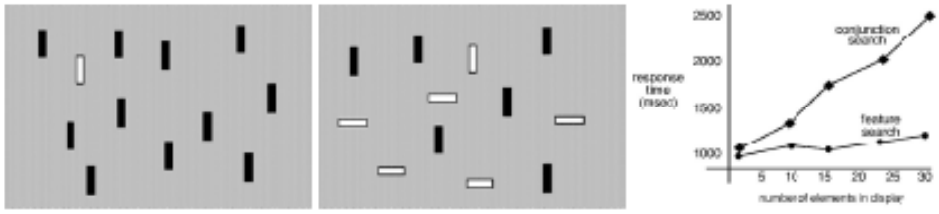
"VISUAL SALIENCY is the distinct subjective perceptual quality which makes some items in the world stand out from their neighbors and immediately grab our attention" (Itti, 2007).

This immediately raises the following questions: How is it that some items in the world stand out more than others and how is this process handled in the human visual system? This question was first addressed in a theory called FEATURE INTEGRATION THEORY which was proposed in 1980 by Treisman and colleagues (Treisman and Gelade, 1980). Using visual search experiments, this theory proposed that only simple visual features – such as color, orientation, size, and spatial frequency – are computed in a parallel way and that the combination of simple features happens – say a red and vertical bar - serialized. Many successful computational models of visual attention are related to this theory.

Hence, it is to be considered as the pioneer in the field of computational modeling of visual attention. Two components of processing visual attention have been suggested in many visual processing studies: 1) Bottom-up processing and 2) Top-down processing.

BOTTOM-UP processing is a fast, mostly unconscious and stimuli-driven computation (Itti, 2003). For example, as illustrated in the following figure, the horizontal red target embedded in a field of horizontal green distractors *pops-out* (cmp. Figure 2.1(a)) and is *salient*. This occurrence is independent of how many distractors in the neighborhood of the target stimuli are present. But not every salient item is as obvious as in a pop-out stimuli. In a visual search task where the distractors are selected similar to the target, before attending the target, one starts first to examine the image. When such stimuli are displayed, it has been shown that subjects' fixations to the attended stimuli takes longer than by for a pop-out stimulus. Targets that require the combination of two features to identify, e.g. red and vertical, are termed CONJUCTIVE TARGET (cmp. Figure 3.1(b)) (Treisman and Gelade, 1980). For conjunctive target search, reaction time increases roughly linearly with the number of distractors. Treisman and Gelade (1980) interpret this as the serial processing of the combination of different features. (e.g. intensity combined with orientation in figure 3.1(b)).

Top-down processing also known as the *task dependent attention* on the other hand guides, the search for a particular target (in memory) to potential targets (for example, a given instruction such as look for the vertical bar). It is a slower processing which directs the attention under volitional control (e.g. personal goal, experience, etc.).



(a) Parallel search task (feature search) for white target among black distractors (pop-out effect).

(b) Serial search task (conjunction search) for a white vertical target among black vertical and white horizontal distractors.

(c) The time subjects need to detect the targets of images (a) and (b).

Figure 2.1 The reaction time (RT) a subject needs to detect a target depends on the complexity of the search task (c). If the target differs only in one feature from the distractors, the search time is almost constant with respect to the number of elements in the display (feature search); the target seems to pop out of the scene (a). If the target is defined by a conjunction of features (conjunction search), the reaction time increases linearly with the number of distractors (b).

2.1.3 Eye Movement and Visual Attention

There exist numerous types of eye movements. The most studied ones are the saccadic eye movements, which are responsible for shifting the high-resolution fovea onto a given target. Once foveated, this target can be processed with more details. Thus, while exploring a given scene

humans shift, successively, their fovea to a set of targets, creating the so called scan path. The question that arises here is how these saccades are generated. As early as 1967, Yarbus ascertained a relationship between saccadic eye movements and visual attention. More recent works argued that visual attention anticipates an eye movement at the same scene location. Furthermore, it has been stated that attention can be shifted about four times before the next eye movement takes place. This behavior allows the attention mechanism to examine several targets and retain the most important one, to which the fovea is then shifted.

Note that the kind of attention deployment that causes eye movement is called overt attention (as opposed to covert attention which is not followed by an eye movement). A large part of the eye saccades is generated and controlled by the visual attention mechanism in order to foveate salient or informative locations of the observed scene. The next relevant question would be how the human visual attention mechanism selects the locations to be attended next. As figure 2.2 shows saccadic eye movements depend on the question subjects were asked. It shows that task has a great effect on the pattern of eye movements. Bottom-up cues are important in free-viewing tasks and in such situations attention is attracted to more salient regions of the scene. However top-down influences come to play role when subjects are doing a specific task or process the image based on their knowledge and motivations. Our aim in this thesis is to model this phenomenon based on a learning approach which considers effect of task in different situations. We use a learning approach to take into account task and visual context.

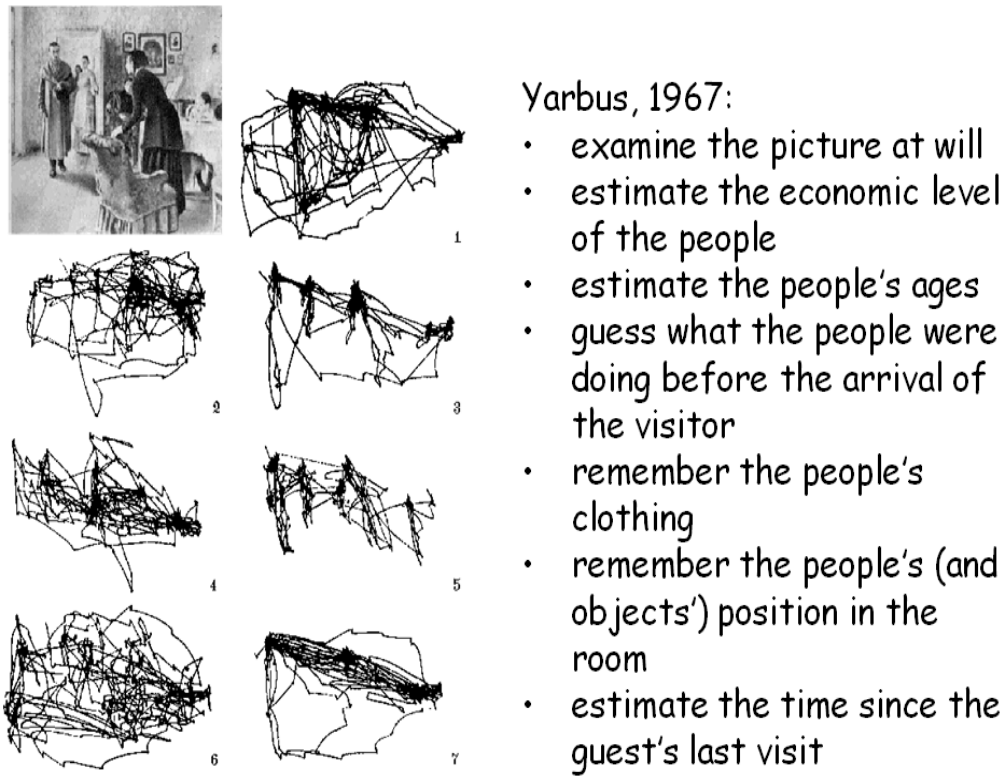


Figure 2.2 Example of a human scan path performed under different questions.

2.2 Computational Models

2.2.1 Introduction

In this section, we will introduce some of the most important computational attention systems, especially those with the highest impact on our work. We start be introducing the model of Koch & Ullman, which laid the theoretical basis for many current attention systems (Koch and Ullman, 1985). Next, one of the currently best-known attention systems is presented: the *Neuromorphic Vision Toolkit (NVT)* of Itti et al. (Itti et al., 1998). It will be described in some detail since it will be used in on our work. Then we explain another work based which has been aimed for modeling task-driven modulations over the saliency model (Navalpakkam et al., 2005). A system which has many features in common with saliency model and has been built upon it is VOCUS and has been presented in (Frintrop et al. 2005). Since this model also enables top-down extension for search task it is related to our work. Finally a model which has been integrated bottom-up saliency model with a hierarchical object recognition model which has the same biological background will be explained (Walther at al. 2006).

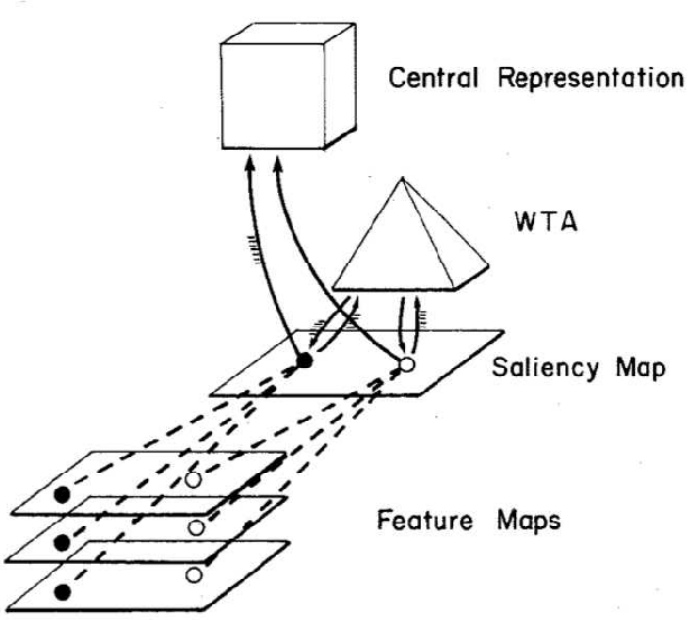


Figure 2.3 The Koch-Ullman model. Different features are computed in parallel and their conspicuities are represented in several feature maps. A central saliency map combines the saliencies of the features and a winner take all network (WTA) determines the most salient location. This region is routed to the central representation where complex processing takes place (Koch and Ullman, 1985).

Koch & Ullman The first approach for a computational architecture of visual attention was introduced by Koch and Ullman (Koch and Ullman, 1985) (see Fig. 2.3).When it was first published, the model was not yet implemented, but it provided the algorithmic reasoning serving as a foundation for later implementations and for many current computational models of visual

attention. The idea is that several features are computed in parallel and their conspicuities are collected in a *saliency map*. A *Winner-Take-All network (WTA)* determines the most salient region in this map, which is finally routed to a *central representation*.

Here, complex processing takes place restricted to the region of interest. The model is based on the *Feature Integration Theory* by Treisman (Treisman and Gelade, 1980): the idea of feature maps that represent in parallel different features as well as the idea of a central map of attention — Treisman's *master map of location* — are adopted. The saliency computations are also influenced by rules called *proximity* and *similarity preferences*, which favor regions that are close or similar to the last focused region. However, newer findings claim that distance has no effect on attentional shifts, that means there is no proximity effect (Remington and Pierce, 1984, Krose and Julesz, 1989). An important contribution of Koch and Ullman's work is the WTA network — a neural network that determines the most salient region in a topographical map — and a detailed description of its implementation. It may be noted that the WTA network shows how the selection of a maximum is implementable by neural networks that means by single units which are only locally connected. This approach is strongly biological motivated and shows how such a mechanism might be realized in the human brain. However, for a technical system a WTA is certainly an overhead since there are much easier ways to compute a maximum from a saliency map. Nevertheless, many computational attention systems take over the idea of a WTA. After selecting the most salient region by the WTA, this region is routed into a *central representation* which at any instant contains only the properties of a single location in the visual scene. Due to this routing, the approach is also referred to as *selective routing model*. How the routing is performed and what happens with the information in the central representation is not mentioned; the idea is that more complex vision tasks are restricted to the selected information.

Finally, the authors suggest a mechanism for inhibiting the selected region causing an automatic shift towards the next most conspicuous location *(inhibition of return (IOR))*. The idea of a central representation in this form is hardly plausible from a biologically point of view: simple and complex processing of visual information in the brain is thought to be more intertwined than suggested by this model. But from a computational point of view the method is suggestive since it enables a modular assembling of different systems: an attentional system for the detection of regions of interest and a recognition system for the detailed investigation of these regions. The proposed architecture is merely bottom-up; it is not discussed how top-down influences from higher brain areas may contribute to the selection of salient regions.

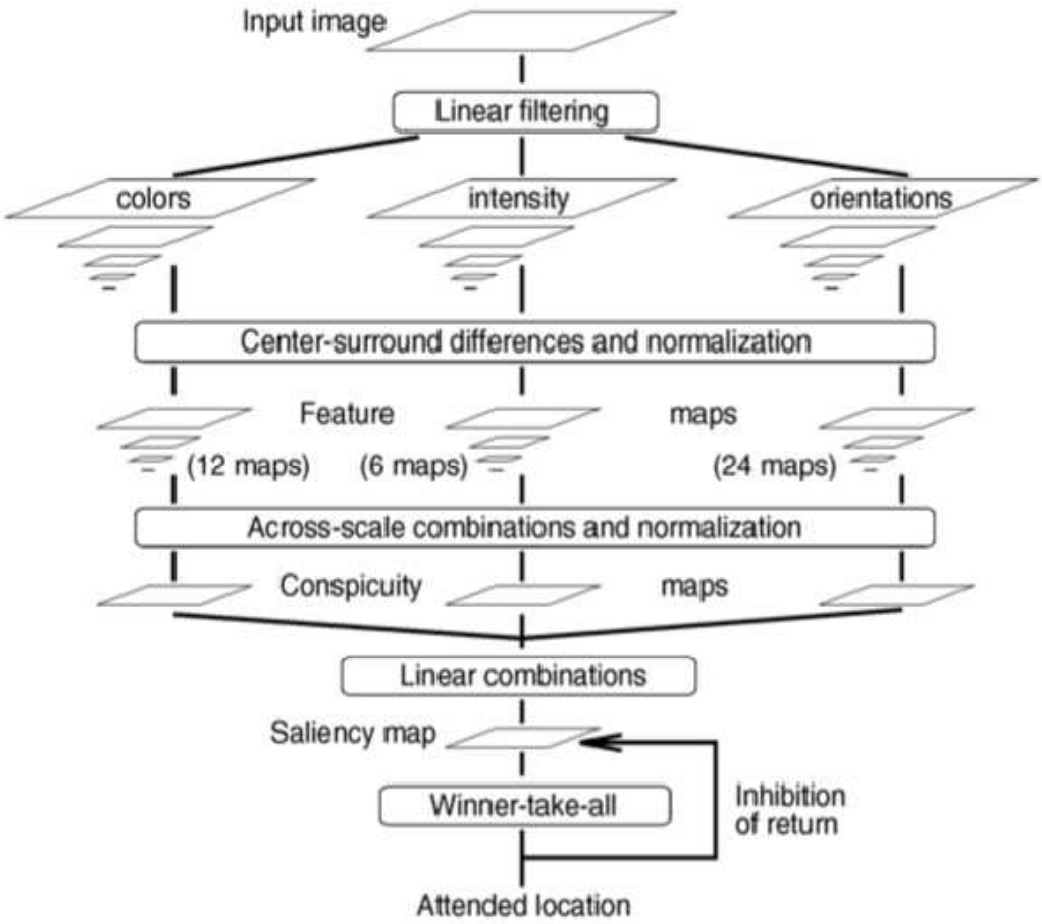


Figure 2.4 Model of the Neuromorphic Vision Toolkit (NVT) by Itti et al. From an input image, three features are computed: color, intensity, and orientation. For each feature, an image pyramid is built to enable computations on different scales. Center-surround mechanisms determine the conspicuities concerning the features which are collected in a central saliency map. A winner take all network determines the most salient location in this map which yields the focus of attention. Inhibition of return inhibits this region in the saliency map and enables the computation of the next focus (Fig. reprinted with permission from (Itti et al., 1998)

Itti. One of the currently best known attention systems is the *Neuromorphic Vision Toolkit (NVT)*, a derivative of the Koch-Ullman model (Koch and Ullman, 1985), that is steadily kept up to date by the group around Laurent Itti (Itti et al., 1998,Itti and Koch, 2001a,Miau et al., 2001,Itti and Koch, 2001b,Navalpakkam et al., 2005). Their model as well as their implementation serves as a basis for many research groups; one reason for this is the good documentation and the availability of the source code for download, allowing other researchers to experiment and further develop the system. Fig. 2.4 shows the basic structure of the model. The ideas of the feature maps, the saliency map, the WTA and the IOR were adopted from the Koch-Ullman Model, the approaches of using linear filters for the computation of the features, of determining contrasts by center-surround differences and the idea of the conspicuity maps were probably adopted from Milanese (Milanese, 1993). The main contributions of this work are detailed elaborations on the realization of theoretical concepts, a concrete implementation of the system and the application to

artificial and real-world scenes. The authors describe in detail how the feature maps for intensity, orientation, and color are computed: all computations are performed on *image pyramids*, *Image pyramid* a common technique in computer vision that enables the detection of features on different scales. Additionally, they propose a weighting function for the weighted combination of the different feature maps by promoting maps with few peaks and suppressing those with many ones. This technique is computationally much faster than the relaxation process of Milanese and yields good results. Since the suggested weighting function still suffered from several drawbacks, they introduced an improved procedure in (Itti and Koch, 2001b).

To evaluate the quality of the NVT, a comparison with human behavior was performed in (Parkhurst et al., 2002). The authors compared how the saliency computed by the system matched with human fixations on the same scenes and found a significant coherence which was highest for the initial fixation. They also found that the coherence was dependent on the kind of scene: for fractals it was higher than for natural scenes. This was explained by the influence of top-down cues in the human processing of natural scenes, an aspect left out in the NVT.

Navalpakkam. The NVT in its basic version does concentrate on computing bottom-up attention. The need for top-down influences is mentioned but not realized. In a recent approach, Navalpakkam and Itti introduce a derivative of their bottom-up model which is able to deal with top-down cues (Navalpakkam et al., 2005). The idea is to learn feature values of a target from a training image in which the target is indicated by a binary mask. Considering the target region as well as a region in the close surrounding — considering 9 locations from a 3×3 grid of fixed size centered at the salient location — the system learns the feature values from the different feature maps on different scales. This yields a 42 component feature vector (red/green, blue/yellow, intensity, and 4 orientations, each on 6 scales). However, it may be doubted if it is useful to learn the scale of a target since during visual search the target should be detected on different scales. During object detection, this feature vector is used to bias the feature maps by multiplying each map with the corresponding weight. Thereby, exciting and inhibiting as well as bottom-up and top-down cues are mixed and directly fused into the resulting saliency map. Figure 2.5 shows this task-driven visual attention model.

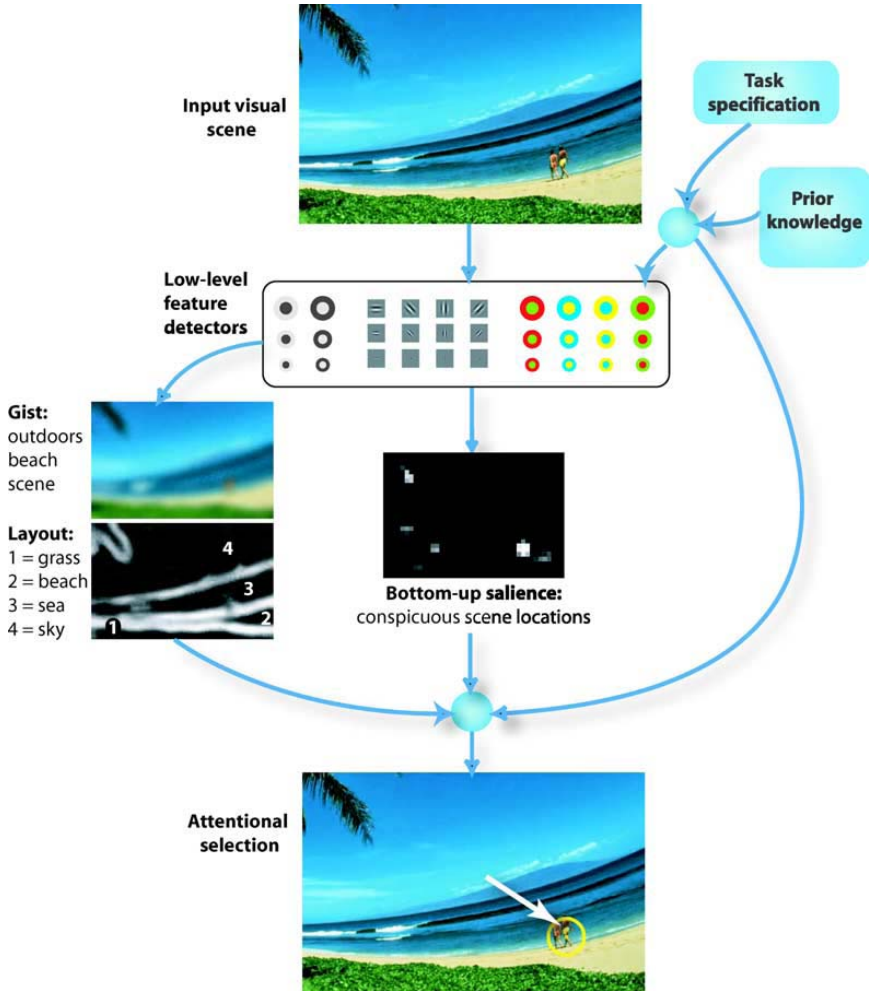


Figure 2.5 Overview of current understanding of how task influences visual attention: Given a task such as "find humans in the scene", prior knowledge of the targets features is known to influence low-level feature extraction by priming the desired features. These low-level features are used to compute the gist and layout of the scene as well as the bottom-up salience of scene locations. Finally, the gist, layout and bottom-up salience map are somehow combined with the task and prior knowledge to guide attention to likely target locations. The present study attempts to cast this fairly vague overview model into a more precise computational framework that can be tested against real visual inputs.

Frintrop. (Frintrop et al. 2005) have introduced a new computational attention system known as VOCUS for efficient and robust detection of regions of interest in images. In their approach, the selection of an image region is determined by two kinds of influences: bottom-up and top-down cues. Bottom-up cues are determined by local contrasts and by uniqueness of a feature. Top-down cues depend on the features of a pre-specified target. Bottom-up and top-down down saliency maps are then linearly added and weighted to form a final saliency map. They have shown that VOCUS is robust and applicable to real-world scenes. Figure 2.6 illustrates this model.

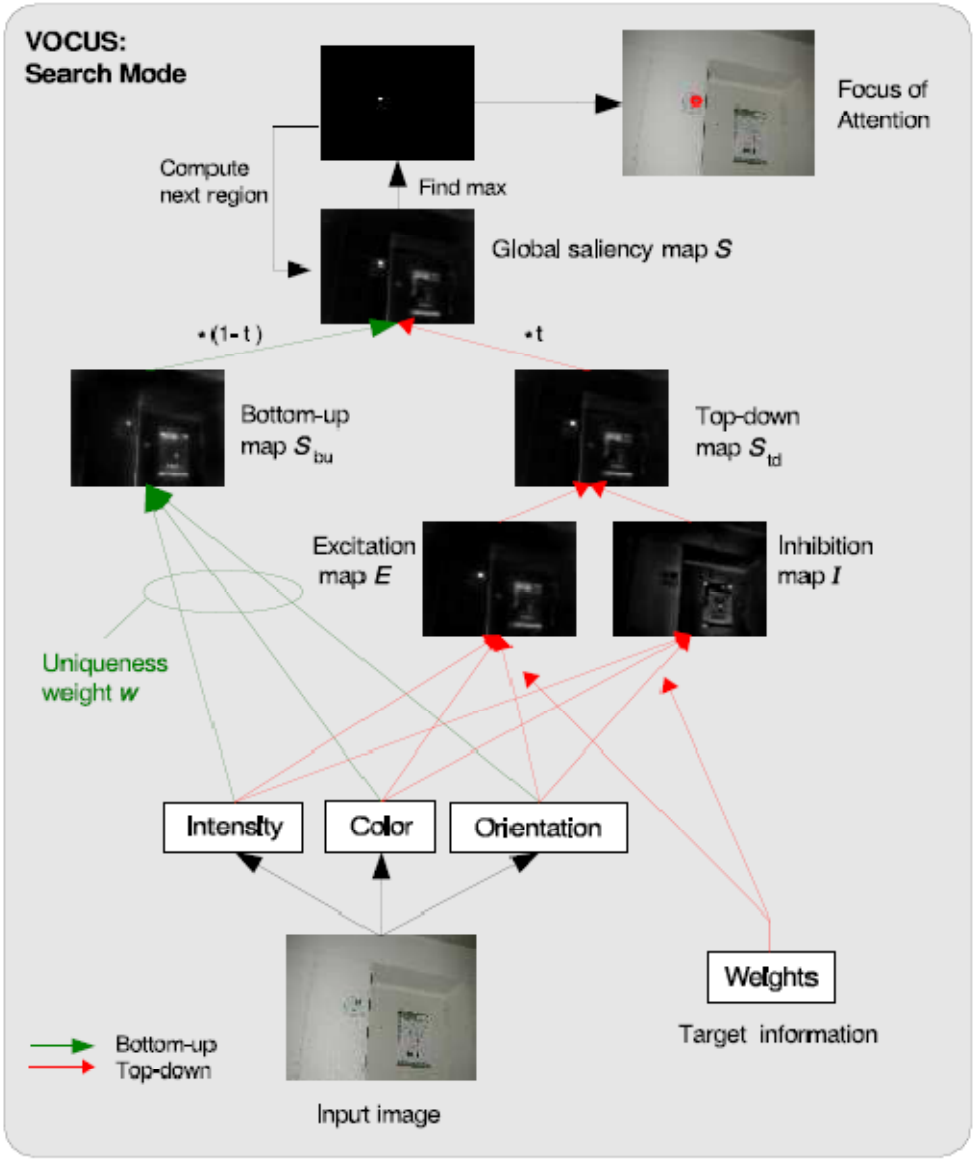


Figure 2.6 VOCUS Attentional System. The bottom-up saliency map S_{bu} competes for saliency with a top-down saliency map S_{tu} which results from an excitation map E and an inhibition map I. These maps result weighted from sum of the feature and conspicuity maps, using the learned vector. When creating the global saliency map S, the influence of bottom-up and top-down is adjustable by the top-down factor t. The images were produced with t=1. (Adopted from Frintrop et al.)

Walther. Miau at al. investigated the combination of the NVT with object recognition, considering in (Miau and Itti, 2001,Miau et al., 2001) the simple biologically plausible recognition system HMAX and in (Miau et al., 2001) the recognition with support vector machines. Walther et al. continued these investigations, starting in (Walther et al., 2002) also with a combination with the HMAX model. In a current approach (Walther et al., 2004), they combine the system with the well-known recognition approach of Lowe (Lowe, 2004) and show how the

detection results are improved by concentrating on regions of interest. Integration of saliency and HMAX models are shown in figure 2.7.

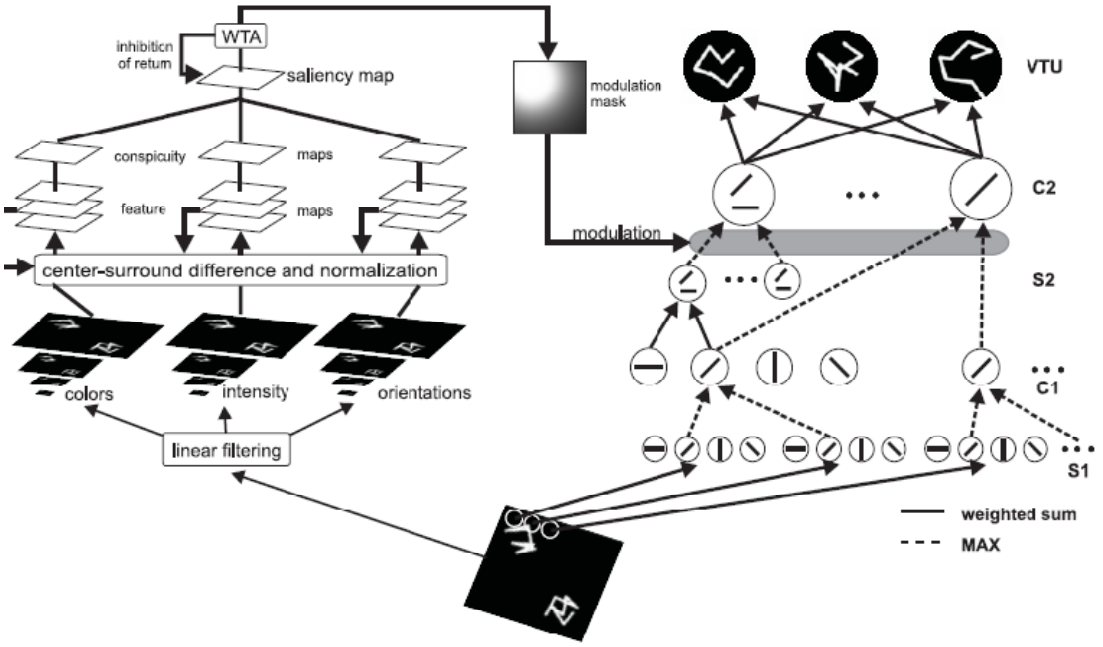


Figure 2.7 Walther's model combines a saliency-based attention system with a hierarchical recognition system. For the attention system, the retinal image is filtered for colors (red-green and blue-yellow), intensities, and four orientations at four different scales, and six center-surround differences are computed for each feature. The resulting 7×6 = 42 feature maps are combined into three conspicuity maps (for color, intensity and orientations), from which one saliency map is computed. All locations within the saliency map compete in a winner-take-all (WTA) network of integrate and fire neurons, and the winning location is attended to. Subsequently, the saliency map is inhibited at the winning location (inhibition-of-return), allowing the competition to go on, so that other locations can be attended to. The hierarchical recognition system starts out from units tuned to bar-like stimuli with small receptive fields, similar to V1 simple cells. In a succession of layers, information is combined alternatively by spatial pooling (using a maximum pooling function) and by feature combination (using a weighted sum operation). View-tuned units at the top of the hierarchy respond to a specific view of an object while showing tolerance to changes in scale and position. The activity of units at level S2 is modulated by the attention system (Fig. 2).

2.3 Applications

While psychological models of visual attention usually aim at describing and better understanding human perception, computational attention systems usually intend to improve technical systems. In this section, we discuss several application scenarios in the field of computer vision and robotics and introduce the approaches that currently exist in this field.

2.3.1 Computer Vision

Object Recognition Probably the most suggesting application of an attention system is object recognition since the two-stage approach of a preprocessing attention system and a classifying recognizer is adapted to human perception (Neisser, 1967). It is worth mentioning that object recognition may be a subtask of more complex applications like object manipulation in robotics, which will be described later. One example of a combination of an attentional front-end with a classifying object recognizer is shown in (Miau and Itti, 2001,Miau et al., 2001). The recognizer is the biologically motivated system HMAX (Riesenhuber and Poggio, 1999). Since this system focuses on simulating processes in human cortex, it is rather restricted in its capabilities and it is only possible to recognize simple artificial objects like circles or rectangles. In (Miau et al., 2001), the authors replace the HMAX system by a support vector machine algorithm to detect pedestrians in natural images. This approach is much more powerful with respect to the recognition rate but still computationally very expensive and lacks real-time abilities. Walther and colleagues combine in (Walther et al., 2004) an attention system with an object recognizer based on SIFT features (Lowe, 2004) and show that the recognition results are improved by the attentional front-end. In (Salah et al., 2002) an attention system is combined with neural networks and an observable Markov model to do handwritten digit recognition and face recognition. In (Ouerhani, 2003), an attention-based traffic sign recognition system is presented.

All of these systems rely only on bottom-up information and therefore on the assumption that the objects of interest are sufficiently salient by themselves. Non-salient objects are not detected and so they are missed. For some object classes like traffic signs which are intentionally designed salient, this works quite well; for other applications, top-down information would be needed to enable the system to focus on the desired objects. It may also be mentioned that when combining object recognition with attention, the advantage over pure classification is usually the time saving and not the quality improvement: most classifiers show no improvement if restricted to a region of interest (an exception is the work of Walther et al. (Walther et al., 2004) since the Lowe detector improves if restricted to a region of interest). Since most attention systems are still rather slow and the recognition systems not powerful enough to deal with a wide variety of objects, the advantage of such a combination of attention and classification does usually not yet show to its best. Currently, there is no existing approach that exhibits a time saving resulting from the combination of attention and classification. However, in future, with more powerful recognition systems and more complex requirements concerning vision systems, an attentional frontend is a promising approach.

A different view on attention for object recognition is presented in (Fritz et al., 2004): an information-theoretic saliency measure is used to determine discriminative regions of interest in objects. The saliency measure is computed by the conditional entropy of estimated posteriors of

the local appearance patterns. That means, regions of an object are considered as salient if they discriminate the object well from other objects in an object data base. A similar approach is presented in (Pessoa and Exel, 1999).

Image Compression A new and interesting application scenario is presented in (Ouerhani et al., 2001): *focused image compression*. Here, a color image compression method adaptively determines the number of bits to be allocated for coding image regions according to their saliency. Regions with high saliency have a higher reconstruction quality with respect to the rest of the image. In (Itti 2005) saliency model is used for image compression. The idea is different coding resolution based on saliency value of image regions.

Image Matching Image matching is the task to redetect a scene, or part of a scene, in a newly presented image. This is often done by matching relevant key points. An approach that uses foci of attention computed by a saliency operator for image matching is presented in (Fraundorfer and Bischof, 2003).

Image Segmentation The automatic segmentation of images into regions usually deals with two major problems: first, setting the starting points for segmentation (seeds) and second, choosing the similarity criterion to segment regions (cf. appendix A.3). Ouerhani et al. present an approach that supports both aspects by visual attention (Ouerhani et al., 2002, Ouerhani and Hugli, 2003a): the saliency spots of the attention system serve as natural candidates for the seeds and the homogeneity criterion is adapted according to the features that discriminate the region to be segmented from its surroundings.

2.3.1 Robotics

Active Vision Active vision represents the technical equivalent for overt attention by directing a camera to interesting scene regions and/or zooming these regions. The goal is to acquire data that is as suitable as possible to the current task and to reduce the processing complexity by actively guiding the sensors (usually the camera) to reasonable regions (Aloimonos et al., 1988). In several cases, active vision is a subtask for applications like human-robot interaction and object manipulation, which will be discussed in the next sections.

In (Mertsching et al., 1999, Bollmann, 1999), the active vision system NAVIS is presented that uses an attention system to guide the gaze. It is evaluated on a fixed stereo camera head as well as on a mobile robot with a monocular camera head. In (Vijayakumar et al., 2001) an attention system is used to guide the gaze of a humanoid robot. The authors consider only one feature, visual flow, which enables the system to attend to moving objects. To simulate the different resolutions of the human eye, two cameras per eye are used: one wide-angle camera for peripheral vision and one narrow-angle camera for foveal vision. Other approaches which use attention systems to direct the gaze of an active vision system are described in (Clark and Ferrier, 1989) and (Driscoll et al., 1998).

Human-Robot Interaction If robots shall interact with humans, it is important that both agree on a current object or region of interest. A computational attention system similar to the human one can help to focus on the same region. Breazeal introduces a robot that shall look at people or toys

(Breazeal, 1999). Although top-down information would be necessary to focus on an object relevant for a certain task, bottom-up information can be useful too if it is combined with other cues. For example, Heidemann et al. combine an attention system with a system that follows the direction of a pointing finger and so can adjust to the region that is pointed at (Heidemann et al., 2004). In (Rae, 2000) this approach is used to guide a robot arm to an object and grasp it. In (Ajalooian et al, 2009) saliency model is used for fast detection of hand regions and then concepts of handmovements are learned for interaction with a robotic marionette.

Object Manipulation in Robotics A robot that has to grasp and manipulate objects first has to detect and possibly also to recognize the object. Attentional mechanisms can be used to support these tasks. For example, Tsotsos et al. present a robot for disabled children that detects toys by the help of attention, moves to a toy and grasps it (Tsotsos et al., 1998). In another approach, Bollmann et al. present a robot that uses the active vision system NAVIS to play at dominoes (Bollmann et al., 1999). The above mentioned approach of Rae in which a robot arm has to grasp an object a human has pointed at, falls also into this category (Rae, 2000).

Robot Localization Another application scenario of an attention system in robotics is the detection of landmarks for localization. Especially in outdoor environments and open areas, the standard methods for localization like matching 2D laser range and sonar scans are likely to fail. Instead, localization by detection of visual landmarks with a known position can be used. Attentional mechanisms can facilitate the search of landmarks during operation by selecting interesting regions in the sensor data. By focusing on these regions and comparing the candidates with trained landmarks, the most probable location can be determined. A project that follows this approach is the ARK project (Nickerson et al., 1998). It relies on hand-coded maps, including the locations of known static obstacles as well as the locations of natural visual landmarks. As already mentioned, (Ouerhani and Hugli, 2004) suggest to use matching and tracking of salient regions for robot localization but a realization of the localization itself has not yet been done.

Chapter 3
State-space Discretization
3.1 Introduction

Standard reinforcement learning explicitly assumes that the agent is able to distinguish between the states of the environment using only its sensors: The perceptual space is said fully observable, and the right decisions can always be made on the basis of the percepts. If it is not the case (i.e. if the perceptual space is only partially observable), the agent cannot distinguish between any pair of states and thus will possibly not be able to take systematically the right decision. This phenomenon is known as the perceptual aliasing (or hidden state) problem, and is closely related to ours. Indeed, Section 4.2 has explained that the incremental selection of a set of perceptual features necessarily leads to a temporary perceptual aliasing, which RLVC tries to get rid of. Early work in reinforcement learning has tackled this general problem in two distinct ways: Either the agent identifies and then avoids states where perceptual aliasing occurs (as in the Lion algorithm (Whitehead and Ballard, 1991), or it tries to build a short-term memory that will allow it to remove the ambiguities on its percepts (as in the predictive distinctions approach (Chrisman, 1992)). Very sketchily, these two algorithms detect the presence of perceptual aliasing through an analysis of the sign of Q-learning updates (cf. Section 2.4.2). The possibility of managing a short-term memory has led to the development of the Partially Observable Markov Decision Processes (POMDP) theory (Kaelbling, et al. 1996), in which the current state is a random variable of the percepts. Hasinoff's technical report (Hasinoff, 2003) is an up-to-date reference about the perceptual aliasing issue.

Although these approaches are closely related to the perceptual aliasing RLVC temporarily introduces, they do not consider the exploitation of perceptual features. Indeed, they tackle a structural problem in a given control task, and, as such, they assume that perceptual aliasing cannot be removed. As a consequence, these approaches are orthogonal to our research interest, since the ambiguities RLVC generates can be removed by further refining the percept classifier. In fact, the techniques above tackle a lack of information inherent to the used sensors, whereas our goal is to handle a surplus of information related to the high redundancy of visual representations.

From another perspective learning top-down attention is highly coupled with learning representations. Therefore the best way to derive visual attention mechanisms is to learn them in concert with visual representations. This is tightly relevant to an area of research known as state space discretization in reinforcement learning. Here we adapt these techniques for deriving spatial visual attention in interactive environments which is like the saccadic eye movements that humans and animals perform to recognize a scene.

Reinforcement learning (RL) is a general framework for modeling the behavior of an agent that learns how to perform its task through its interactions with the environment. The only information that the agent takes in response to its actions is a reinforcement signal instead of being told that its action has been right or wrong. Like many existing learning methods, RL suffers from the curse of dimensionality, requiring a large number of learning trials as state-space grows. But it has the ability to handle dynamic and non-deterministic environments. It is believed that curse of dimensionality can be lessened to a great extent by implementation of state abstraction methods and hierarchical structures. Moreover, incremental improvement of agent's performance becomes much simpler due to less number of states.

Several approaches for interactive discretization of state space have been proposed. Techniques using a non-uniform discretization are referred to as variable resolution techniques (Munos & Moore, 2002). The parti-game algorithm (Moore & Atkeson, 1995) is an algorithm for automatically generating a variable resolution discretization of a continuous, deterministic domain based on observed data. This algorithm uses a greedy local controller and moves within a state or between adjacent states in the discretization. When the greedy controller fails, the resolution of the discretization is increased in that state. The G algorithm (Chapman & Kaelbling, 1991), and McCallum's U-Tree algorithm (1996), are similar algorithms that automatically generate a variable resolution discretization by re-discretizing propositional techniques. Like parti-game, they both start with the world as a single state and recursively split it when necessary. The continuous U-Tree algorithm described in (Uther & Velso, 1998), extends these algorithms to work with continuous state spaces.

Jodogne et al. (2007) have proposed an approach for learning action-based image classification known as Reinforcement Learning of Visual Classes (RLVC). RLVC consists of two interleaved learning processes: an RL unit which learns image to action mappings and an image classifier which incrementally learns to distinguish visual classes. RLVC could be regarded as a feature based attention method in which the entire image has to be processed to find out whether a specific visual feature exists or not in order to move in a binary decision tree. Like RLVC, our approach also extends the U-Tree to visual domain. Our approach tackles this weakness of the RLVC, the exhaustive search for a local descriptor over the entire image, by computing and searching local descriptors at few spatial locations. Once aliasing is detected a spatial location is selected which reduces the perceptual aliasing the most. This local visual processing results in very fast scene recognition because features are extracted in small regions and there is no need for exhaustive search for a feature.

3.2 RL

In RL (Sutton & Barto, 1998), the environment is modeled as a Markov Decision Process (MDP). An MDP is a 4-tuple $\langle S, A, \mathcal{T}, \mathcal{R} \rangle$, where S and A are finite sets of states and actions respectively, $\mathcal{T}$ is a probabilistic transition function from S×A to S, and $\mathcal{R}$ is a reinforcement function from S×A to $\mathbb{R}$. An MDP obeys the following discrete-time dynamics. If at time t, the agent takes the action at, while the environment lies in state st, the agent perceives a numerical reinforcement $r_{t+1} = \mathcal{R}(s_t, a_t)$, then reaches some state st+1 with probability $\mathcal{T}(s_t, a_t, s_{t+1})$. For an infinite sequence of interactions starting in a state st, $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1}$ is the discounted

return where $\gamma \in [0,1]$ is the discount factor that gives the current value of the future reinforcements. The solution to the Markovian Decision Problem is to find an optimal percept-to-action mapping that maximizes the expected discounted return. Q-learning is a direct, interactive, and adaptive solution (Watkins & Dayan, 1992) with the convergence proof. It is used here for learning top-down attention in concert with motor actions.

3.3 U-TREE

The U-Tree (McCallum, 1996) abstracts the state space incrementally. Each leaf l_t of the U-Tree corresponds to an abstract state s_t. Leaves store the action-values $\boldsymbol{Q(s_t, a_t)}$ for all available actions a_t. The tree is initialized with a single leaf, and new abstract states are added if necessary. Sub-trees of the tree represent subtasks of the whole task. Each sub-tree can have other sub-sub-trees that correspond to its sub-sub-tasks. The hierarchy breaks down to the leaves that specify the primitive sub-tasks. The procedure for construction of the abstraction tree loops through a two phase process: *sampling* and *processing*. During the sampling phase the algorithm behaves as a standard RL, with the added step of using the tree to translate sensory input to an abstract state. A history of the transition steps, i.e. $\boldsymbol{T_t = (s_t, a_t, r_t, s_{t+1})}$ composed of the current state, the selected action, the received immediate reward, and the next state is recorded. The sample is assigned to a unique leaf based on the value of the current state. Each leaf has a list per action for recording the sampled data-points. After some learning episodes the processing phase starts. In this phase a value is assigned to each sample:

$$\boldsymbol{V(T_t) = r_t + \gamma V(\bar{s}_{t+1}), V(\bar{s}_{t+1}) = \max_a Q(\bar{s}_{t+1}, a)} \qquad (\mathbf{2.1})$$

where $\boldsymbol{\bar{s}_{t+1}}$ is the abstract state that $\boldsymbol{s_{t+1}}$ belongs to. If a significant difference among the distribution of sample-values within a leaf is found, the leaf is broken up to two leaves. To find the best split point, the algorithm loops over the features. The samples within a leaf are sorted according to a feature, and a trial split is virtually added between consecutive pairs of the feature values. This split divides the abstract state into two sub-sets. A splitting criterion compares the two sub-sets, and returns a number indicating the difference between their distributions. If the largest difference among all features is bigger than a confidence threshold, then the split is introduced.

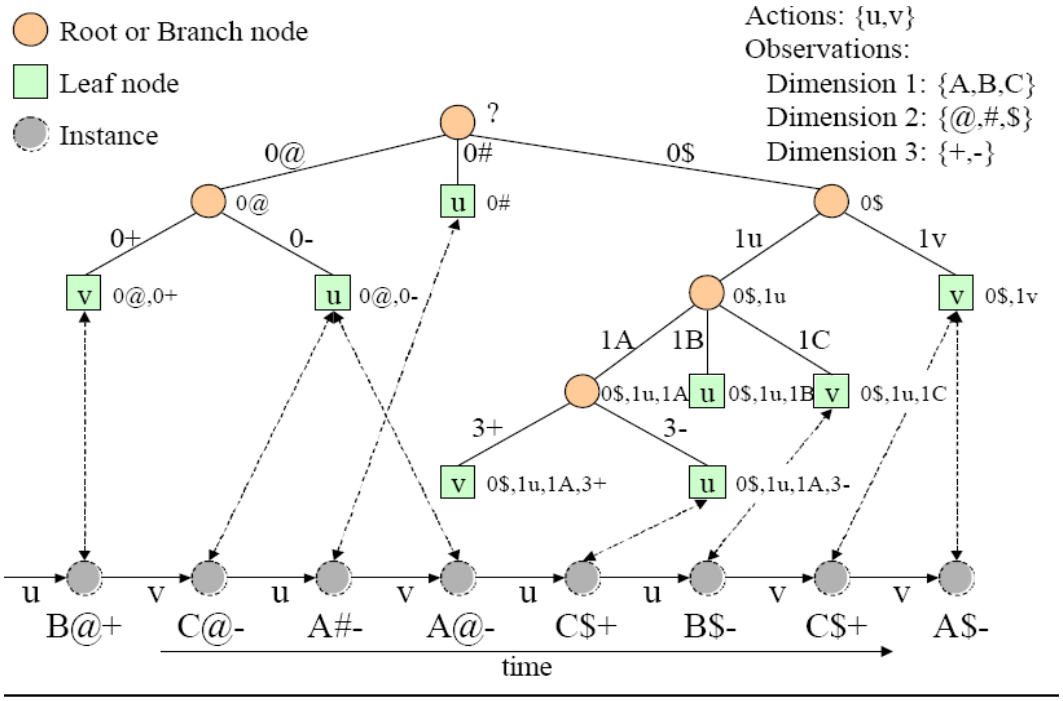


Figure 3.1 A U-Tree instance-chain and tree. The instance-chain is at the bottom: each instance circle is labeled by the observation received at that time step, each instance transition is labeled by the action taken to make that transition. In the tree, each branch is labeled by its history count and perceptual dimension label; beside each node is the conjunction of features the node represents. The tree nodes drawn as squares are the agent's internal states; each contains a Q-value for each action, although the figure only shows the policy action. The dashed-arrows show which agent internal state holds each instance. Adopted from (McCallum et al.)

3.4 RLVC

RLVC (Jodogne & Piater, 2007) shares the same theoretical basis as the original U-Tree algorithm. It iteratively builds a series of classifiers $\boldsymbol{C} = \{\boldsymbol{C_1}, \boldsymbol{C_2}, \dots, \boldsymbol{C_k}\}$ that discretizes the perceptual space *P* into a finite set of perceptual classes by testing the presence of features in the percepts. After *k* refinement steps, the classifier denoted by C_k, partitions the visual perceptual space *P* into a finite number m_k of *visual classes* $\{\boldsymbol{v_1^k}, \dots, \boldsymbol{v_{m_k}^k}\}$. C_k is a binary decision tree, each internal node of which tests the presence of a given visual feature, and each leaf of which corresponds to a visual class. In iteration *k,* a visual class, $\boldsymbol{v_i^k}$, is refined if it has perceptual aliasing meaning that optimal decisions cannot be always made since different percepts requiring different actions are grouped together. In order to reduce aliasing, visual classes with aliasing are expanded by checking the existence of a feature. To detect nodes with aliasing *Bellman's Residual* or *TD-error* is measured for all experiences under a visual class which is defined as:

$$\boldsymbol{B_k(s,a) = \mathcal{R}(s,a) + \gamma \max_{a' \epsilon A} Q_k'^{*}\left(C_k\left(\mathcal{T}(s,a)\right), a'\right) - Q_k'^{*}(C_k(s), a)} \qquad (2.2)$$

where $Q_k^{'*}$ and B_k are the estimation of the optimal Q function using C_k and the residual at state s, respectively. A non-zero $B_k(s,a)$ for some time stamp t indicates the presence of aliasing in the visual class C_k (s) with respect to the action a. After aliasing is detected a SIFT feature (Lowe, 2004) from the percepts in C_k (s) is selected which mostly reduces the variance of Bellman residuals and also results in two significantly different distributions. Figure 3.2 shows the structure of the RLVC algorithm.

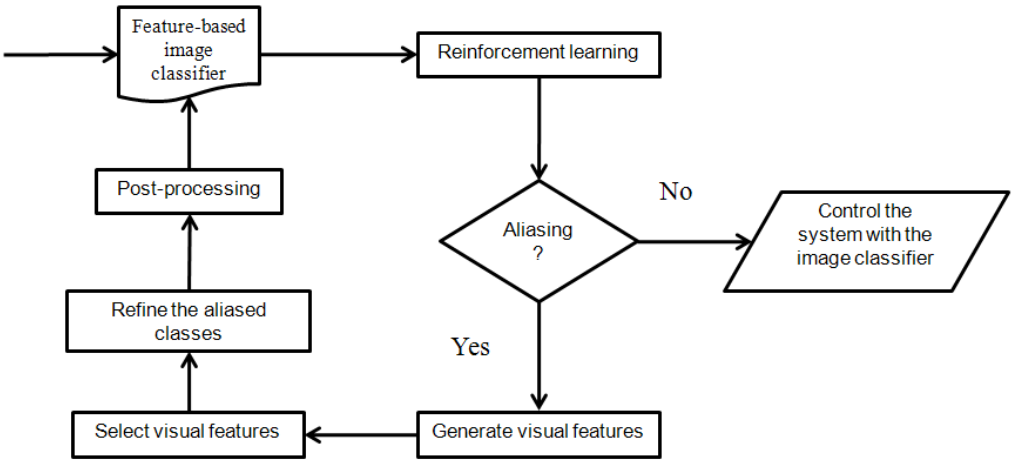


Figure 3.2 The different components of the RLVC algorithm.

An example application of the RLVC over the visual Gridworld shown in Figure 3.3 is shown in Figure 2.9. As shown by Figure 3.4, RLVC succeeded to learn the optimal policy by discriminating among objects in the grid. Shown binary tree checks SIFT features in order to find the class of an object.

Figure 3.3 Small visual Gridworld topology. Cells with a cross are walls, and the exit is indicated by a gray background. Empty cells are labeled by a picture, in which circles indicate the interest points that are detected by the Color Harris detector.

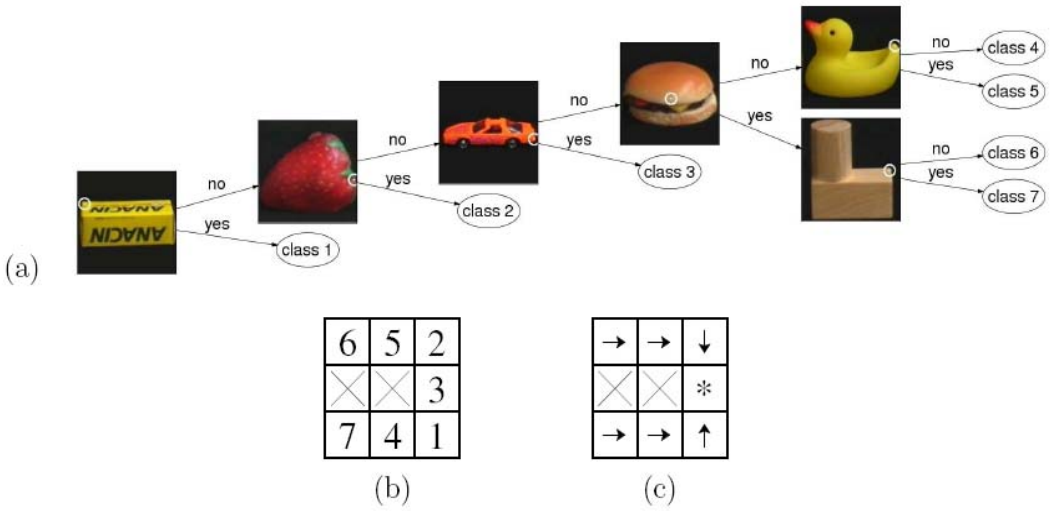


Figure 3.4 Resolution of the small visual Gridworld by RLVC: (a) The final classifier Ck that tests the presence of the circled local-appearance features, (b) the label of the perceptual class that is assigned to each empty cell by Ck, and (c) the computed optimal policy for this classification.

Part II
Learning Top-down Attention Control

Chapter 4
Associative Learning of Attention and Action

Ali Borji, Majid Nili Ahmadabadi, Babak Nadjar Araabi

Published in:
IROS2008 workshop on from motor to interaction learning in robots

Abstract. Like humans and primates, artificial creatures like robots are limited in terms of allocation of their resources to huge sensory and perceptual information. Thus attention is regarded as the same solution as humans in this domain. While bottom-up attention is determined by the image statistics, top down attention is dependent the on behavior and the task an agent is doing. This work attempts to consider a task based top-down visual attention control when resources of the agent are limited. Particularly attention control is formulated as an optimization problem in which the agent has to gain maximum reward while satisfying a constraint which is its information processing bottleneck. Reinforcement learning is then used to solve that optimization problem. A driving environment is simulated in that agent has to learn how to drive safely by attending to the right spatial locations and performing appropriate motor actions.

Key words: Reinforcement Learning, Q-Learning, Spatial Attention

4.1 Introduction

Possible confusion because of huge sensory space, limited response time, dynamicity of perceptual space and environment, and accuracy and reliability of sensors causes a bounded rationality for an agent instead of full rationality. In this regard attention control could act like enhancing the rationality of the agent by proposing small amount of information for further processing. For the agent to be successful in terms of receiving maximum cumulative reward, it should be able to perform perceptual actions or attentions as well as motor actions at the same time. These perceptual actions are available at several forms like where to look and what to look in visual modality. In situations where the environment is unfamiliar or not clearly defined or there is a small training dataset, fixed design of a control strategy is helpless. Therefore interactive learning methods like reinforcement learning are the first options in these cases.

In order to better understand the mechanisms of attention, it is not just enough to examine simple synthetic or natural stimuli in laboratory because attention behavior under top down control is mostly determined by task demands and characteristics which change through time in

spatial and temporal manner. To better cope with this type of attention, we should consider embodiment and situatedness which are two main concepts of new trends in artificial intelligence (Clark & Grush, 1999). The main claim behind this concept is that behaviors like attention, emotion, etc evolve depending body characteristics (dynamics, form, etc) and the environment which agents live in. Thus attention could not be studied and understood in an abstract manner. Therefore the study of attention is much influenced by actions which are based on visual information like grasping, etc. For instance to drive safely needs very complex behaviors like detecting and monitoring stop signs, traffic lights, other cars as well coordinating actions and attentions. That's why novice drivers usually consider driving in crowded big cities hard and challenging.

In this work we consider attention control as an optimization problem. In this optimization framework an artificial agent has a certain limited amount of processing power which he should devote them to the most appropriate items from the sensory space. Particularly our aim is to propose a mechanism for learning top top-down visual attention control for an agent performing tasks which needs motor actions in natural interactive environments like driving. Both top down attention control signals and motor actions will be learned concurrently to fulfill task demands. Our approach is motivated by studies in neuroscience and psychology of attention.

4.2 Related Works

A broad range of works in modeling visual attention are reported in the literature. In this section we review those works which are related to ours, especially those which have considered learning aspects of attention control. Unfortunately, there are a few researches on learning and formation of attention control; rather they are mostly related to the attention modeling. Our focus will be to review those works which has followed a learning based approach to develop visual routines specially attention. In this regard we look at those works which have merged those two topics.

One of the earliest studies which showed that attention is task based and is controlled top-down is done by Yarbus (Yarbus, 1967). He showed that task has a great effect on the pattern of eye movements. Bottom-up cues are important in free-viewing tasks and in such situations attention is attracted to more salient regions of the scene. However top-down influences come to play role when subjects are doing a specific task or process the scene based on their knowledge and motivations. Some previous works have tried to implement top-down influences. One drawback of those approaches is that they have tackled the problem in a very limited and abstract way and have not considered the temporal aspects of top-cues such as task demands on attention which happens in our every day behavior. In brain these processes are done mainly in frontal cortex where short and long term memories are stored and our intentions and plans are formed. It is clear that attention is not such an episodic process but is also dependent on temporal manner the stimuli are presented and also temporal needs and internal states of the agent. This approach has been followed up in active vision systems for building robots able to control their gazes according to their tasks. One of the outstanding works on visual attention which is based on an earlier model of visual attention developed by Koch and Ullman (Koch & Ullman, 1998) is proposed in (Peters & Itti, 2008). Simplicity and little processing time are two main advantages of this model. It is continuously updated and has been the basis of many other models. It's called saliency based model and have been used to explain behavioral data on simple synthetic static

search arrays to dynamic natural stimuli like movies and interactive ones like game playing (Peters & Itti, 2008).

In (Reichle & Patryk, 2006), a reinforcement learning method is proposed to model the reading task. A model for mimicking an expert reader is introduced by predicting where eyes should look (spatial attention) and how long they should stop there in order to lead to the best comprehension from the text. This is an interesting work which proposes an optimization framework for attention and then uses RL to solve it. Our approach is the same, but with the difference that we are aim to apply it to complex tasks where agents have to perform physical actions while controlling their sensory acquisition behavior. Furthermore this research has the limited generalization and could only explain reading experiments.

In (Fritz et. al, 2004, Paletta et al., 2005), a 3-step-architecture is presented which first extracts attention focus according to information theoretic saliency measures. Then, by searching in pre-specified areas from the first step, decides whether the object is available in the image or not and finally suggests a shift for attention. The final step is done using Q-learning with the goal to find the best perceptual action according to the search task. This research is related to our work because it also couples decision making and attention control and uses a reinforcement based learning approach. But, due to the simple task of search in a limited image database, the approach is very simple and the application domain is limited.

In (Gonic et al., 1999), an approach for attention control is presented in a robotic platform with neck, eyes and arms. The first approach is a simple feed forward method which uses back-propagation learning algorithm while the second one uses reinforcement learning and a finite state machine for state space representation. The robot has 3 types of actions: attention shift, visual improvement and haptic improvement. Their results confirm that the second approach generates a better performance in terms of finding previously observed objects even with fewer movements in head and neck and also shifting the attention.

In (Sprague & Ballard, 2005), a new model for human eye movements is proposed that directly ties eye movements to the ongoing demands of behavior. The basic idea is that eye movements serve to reduce uncertainty about environmental variables that are task relevant. A value is assigned to an eye movement by estimating the expected cost of the uncertainty that will result if the movement is not made. If there are several candidate eye movements, the one with the highest expected value is chosen. The model is illustrated using a humanoid graphical figure that navigates on a sidewalk in a virtual urban environment.

An approach for learning gaze control for a mobile robot is proposed in (Minut & Mahadaven, 2001), which proposes a model of selective attention for visual search tasks. They have implemented their model using a fixed pan-tilt-zoom camera in a visually cluttered lab environment, which samples the environment at discrete time steps. The agent has to decide where to fixate next purely based on visual information, in order to reach the region where a target object is most likely to be found. The model consists of two regions in the room for reaching the target object, using as objective function which is the expected value of the sum of discounted rewards. By selecting an appropriate gaze direction at each step, this module provides top-down control in the selection of the next fixation point. The second module performs “within fixation” processing, based exclusively on visual information. An interesting point with this work is that it has incorporated learning where to look in a simple visual search task. Main feature of this work is its implementation on a real robotic agent working in a natural environment.

A study reported in (Maljkovic & Nakayama, 1994), supports the idea that attention could be learned by past experience. In a behavioral task authors observed that subjects showed better performance in successive trails, which evidently is in favor of learning attention control. A modeling work trying to explain such data is done in (Mozer et al., 2006). They have proposed an optimization framework to minimize an objective function which is the sum over reaction time of each state weighted by the probability of that state to occur. Then by means of a Bayesian belief network they minimized this objective function.

A biologically motivated framework for learning attention control and decision making is reported in (Shariatpanahi and Ahmadabadi, 2007). Physical actions and attentions are integrated in augmented actions and reinforcement learning is used to solve an optimization problem which is to maximize the cumulated reward. Mirror neurons are proposed for concept extraction and knowledge abstraction to deliver to other environments and tasks. Our work follows the integration of attention control and action selection for an artificial agent performing a complex cognitive task, but with the distinction of adding rich visual sensory information, achieving complex tasks in natural interactive environments, coping with biology and incorporating other sources which influence visual attention.

In regard with the above researches, we consider attention control and decision making in a unified framework and formularize them as an optimization problem where agent has a limited number of processing resources while facing a lot of sensory information. Then by incorporating reinforcement learning, we try to solve that optimization problem. Some advantages of such a learning approach are easy generalization, knowledge transfer and no need for exact modeling of the environment in advance.

4.3 Proposed Approach

Agent is to learn performing physical actions and visual attentions simultaneously in a working environment in order to optimize an objective (goal) function. Agent captures visual information from the scene in the form of still images or video through its visual sensor. Captured visual scene undergoes an early visual processing. This stage contains two units which act in parallel. According to psychophysical findings, we are not blind outside our focus of attention, and information outside the focus of attention is also processed but not to the extent of the attended area. A bottom-up visual attention unit selects a subset of image and GIST extractor performs a rough categorization of the image (Olica & Torralba, 2006).

Visual information from attended area, visual sensor and GIST extractor enters to the higher visual processing unit. This unit is aimed to derive cognitive information useful for decision making (eg. scene interpretation, object recognition, etc). Processing in this unit are dependent on the environment the agent is living and the type of tasks it is supposed to do. Although other higher processes like emotions, internal events, etc are also important in determination of visual attention, we limit ourselves here to those cues produced only based on visual scene and task demands.

Next, based on this information a decision has to be made. A state extractor unit in decision making state gets input from the higher cognitive unit as well as GIST extractor which determines spatial context, and outputs a well defined state. Reinforcement learning is used to find the best policy which optimizes an objective function. An augmented action is generated which consists of a motor action and perceptual action (top-down attention signal). A reinforcement signal which

is the effect of the motor action and implicitly attention is fed back to RL unit by a critic, which evaluates this action. Reinforcement learning is used here because of its flexibility, power where critic's information is limited as well as its biological plausibility.

Theoretically any bottom-up visual attention unit could be used in early visual processing stage. For example to extend the model to work with natural scenes, saliency based model of visual attention could be used. Overall model is summarized in Figure 4.1.

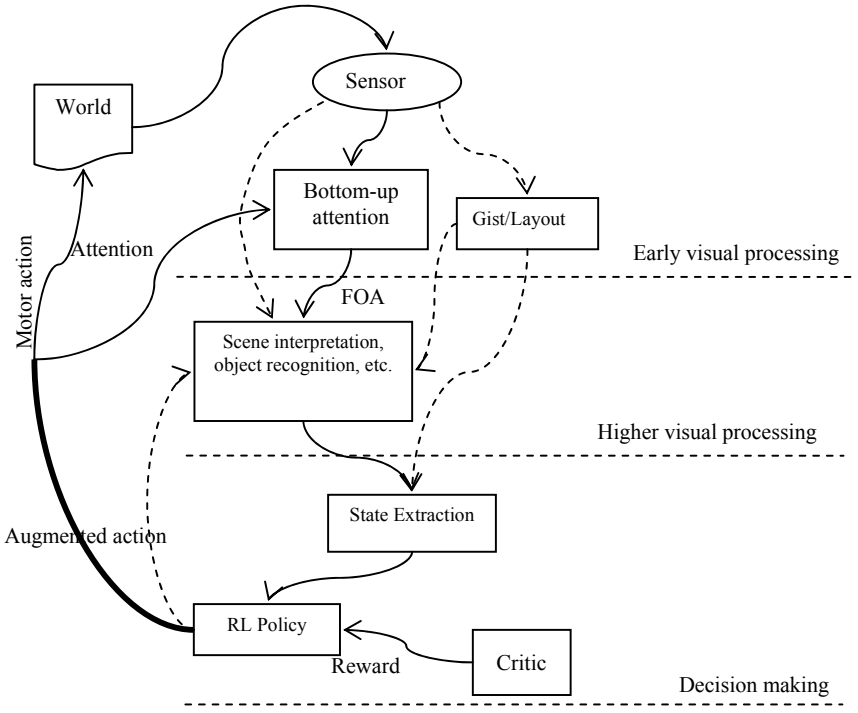


Figure 4.1 Agent is doing a behavior or a task in it. A visual stimulus is captured by the visual sensor of the agent. This stimulus is processed by an early visual system which could be artificial counterpart of V1 and V2 of the primary visual cortex and also by a bottom-up visual attention system. A saliency map is the output of this unit. Evidences exist for availability of such a saliency map in multiple brain areas like FEF, SC, LIP and V4. Another top-down component is based on the idea of "Gist," which in psychophysical terms is the ability of people to roughly describe the type and overall layout of an image after only a very brief presentation, and to use this information to guide subsequent target searches. Higher cognitive processes which are believed to occur in higher visual areas V4/IT perform a variety of cognitive visual processes like scene understanding, object recognition, etc. Conceptual output of this unit is transferred to prefrontal, posterior parietal, LIP, etc which perform a sensory motor processing and action selection. Two control signals (motor action and attention) are generated which affect the world and bottom-up visual attention system. A critic assigns a reinforcement signal which evaluates the actions. Dashed lines in the figure are not implemented yet.

4.4 Mathematical Formulation

Top-down attention is represented as a vector of n elements:

$$\overline{ATT} = \{\overline{att}_1, \overline{att}_2, \ldots \overline{att}_n\} \tag{4.1}$$

with each $\overline{att_i}$ being the parameter vector of the bottom-up visual attention system. In general extracting area of interest (attended area) is a function *f* which takes an image I and a top-down attention $\overline{ATT}$ and outputs a salient region:

$$\overline{AOI} = f(I, \overline{Attention}) \tag{4.2}$$

Higher level processing is modeled as a function g which takes attended area (AOI) and GIST as input and outputs a state S (higher visual information and information about scene):

$$S = g(\overline{AOI}, GIST, I) \tag{4.3}$$

S is a cognitive representation of the scene rich enough to help the agent make decisions. If necessary the scene representation should be transformed into a state by another mapping. After these steps, an RL unit produces an augmented action a, which is composed of a motor action m and a perceptual action b. This attention vector biases the bottom up model of attention. This command, could act as an explicit spatial shift toward an object or a position with high value in the saliency map, a shift in weightings of different features or modalities, modulation of a region in the visual field, etc. State of the agent is its subjective representation of the external world and itself at every decision making point. This representation should be rich enough to support bounded rational decision making.

In learning component (RL), attention control and motor responses are optimized gradually and interactively. This module is responsible for learning optimal decisions at each state S. The optimal decision is found using an RL method which seeks a decision making strategy (policy) at each state that maximizes agent's expected return. In our model we use discounted expected reward as expected return and Q-learning because of its biological plausibility.

A(s) is the set of all possible actions in each state, M(s) is the set of motor actions and B(s) is the set of perceptual actions. Null is added to both types of actions because in some situations no action or attention is needed.

$$A(s) = \{M(S) \cup Null\} \times \{B(S) \cup Null\} \tag{4.4}$$

For agent to find its optimum action selection policy π(s, a) to maximize the expected value of the reward signals r it receives, uses a *Q*-learning algorithm. The value of taking action a in state s under a policy π is denoted by *Q(s, a)* and is updated in each time stec
p according to following equation:

$$Q(S_t, a_t) = (1 - \alpha)Q(S_t, a_t) + \alpha\left(r_t + \gamma \max_a Q(S_{t-1}, a)\right) \tag{4.5}$$

in above formula, α is the learning rate and γ is the discount factor, *π(s,a)* (probability of taking action a given state s) is then determined by the values of Q*π(s,a)* in a softmax or ε-greedy manner. Algorithm of the discussed attention control system is summarized in table 1.

Table 4.1 Algorithm for learning top-down visual attention control

Algorithm	
1	Get a visual image from the visual sensor I
2	Apply the top-down attention signal *b* over the input image (bias the bottom up model) and report the attended area AOI
3	Perform a higher cognitive processing over the attended area and derive a state s.
4	Based on policy π(s), select an action a = (m, b), m ε M, b ε B.
5	Get a reward r from the critic.
6	Update the current policy π(s) by updating the value of Q(s, a) by equation x.
7	If learning is not converged go to step one
8	Stop.

4.5 Experiments and Results

In order investigate the efficiency of the proposed method, we applied it to a simulated driving experiment. In this experiment, agent tries to discover rules and spatial locations of interest of an unknown environment by performing actions and attentions until it learn a set of best actions and attentions in each state. A mobile robot with a limited field of view (FOV) is supposed to learn how to drive in a simulated driving site and extract the rules which are set by the designer in advance. Environment is a simulated driving world with crosses, three ways, turns, etc. Like real world situation some rules for driving are imposed in this environment which if driving agent ignores them will receive punishments. In Figure 4.2 visual field of view of the agent is shown. Visual sensor of the agent captures a scene from this field. The set of location that agent (its field view) sees at any moment are:

$$\overline{FOV} = \{forward\ left\ (FL), far\ forward\ (FF), forward\ right\ (FR), \\ left\ (L), forward(F), right(R)\} \quad (4.6)$$

This large set of visual data makes an attention mechanism more necessary and useful. As it is assumed that one location could be attended in any moment because of agent's limited processing power. Attention and motor actions the agent could perform are as follows:

$$\overline{Attention} = \{FL, FF, FR, L, F, R, Null\} \quad (7)$$

$$\overline{Action} = \{go\ forward(F), turn\ right(TR), turn\ left\ (TL), Null\} \quad (4.7)$$

Each cell in the grid is represented by '0' if it is open (white cell making the way) and '1' if it is occupied. A state is represented by the contents of the cells it observes augmented by another value which is the content of the cell (value of the sign) at the attended location.

$$\overline{State} = \{FL, FF, FR, L, F, R, sign\ value\ at\ attended\ location\} \quad (4.8)$$

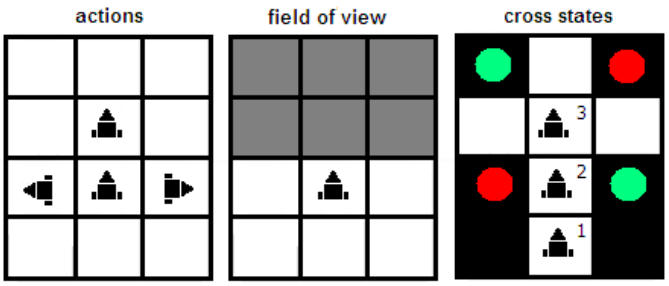


Figure 4.2 Motor actions of the agent are shown at the left. Agent at any moment could choose to either turn left or right, go ahead or stay at its current position. Turning moves the agent one block to the direction of turn. At right is shown field of view of robot (FOV). Agent sees all locations in its visual field but could only attend to one location because of limited processing power. Right side shows the states of the agent behind a cross represented at table 2. In four sides of the cross are signs which alarm the status of the sign. Agent must learn to attend to the sign at its forward right and perform the right action in each situation. Signs are randomly changed in our simulation to generate all possible of states.

A '9' at the end of a state means the value of the sign is not relevant and does not change the state of the agent. For example, consider the situation of the robot behind a cross shown in the right side of Figure 4.2. Moving from position one to position two, agent's state changes to 1010004 provided that attention is FR and light is red which is coded here by a '4' and 1010005 if attention is FR and light is green. Driver must attend to the light in its forward right position and stay behind the cross if the light is red and pass if it gets green.

Table 4.2 States of the robot behind the cross and their right actions and attentions that must be performed as well as their associated rewards at Figure 4.2. Agents was also punished (r=-2) whenever it went off-road. 'Rep.' stands for 'Representation'.

Rule	Representation	Action	Attention	Reward	Comment
1	0001019	F	FR	+3	Position 1 behind cross
2	1010002	N	FR	+3	Position 2 behind cross, red light
3	1010003	F	N	+2	Position 2 behind cross, green light
4	1011012	F	N	+2	Position 3 in cross, red light
5	1011013	F	N	+2	Position 3 in cross, green light

Some of the simulated driving environments are illustrated in Figure 4.3. Figure 4.4 shows the learning curves of a sample run.

4.6 Conclusions

In this paper, we proposed top-down model of attention control that is in cross roads of neuroscience and engineering. For that, we considered findings from both artificial intelligence

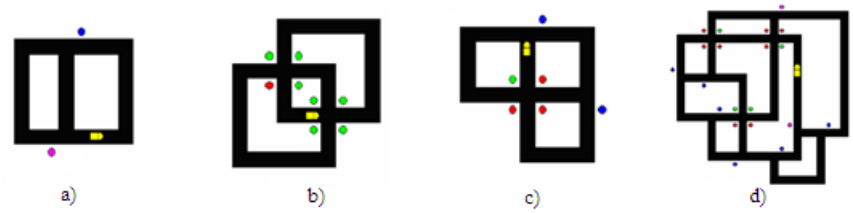


Figure 4.3 Examples of simulated driving environments from simple to complex (a to d). Environment could have crosses, turns, three ways etc. Rules for attention and action are associated with each state. Agent has to learn such rules in order to gain maximum reward. Lights were randomly changed in simulation in order to produce all combinations of states. Agent could learn in simple environments and later benefit from that knowledge in more complex ones by transferring its knowledge or it could start by learning in the complex environments.

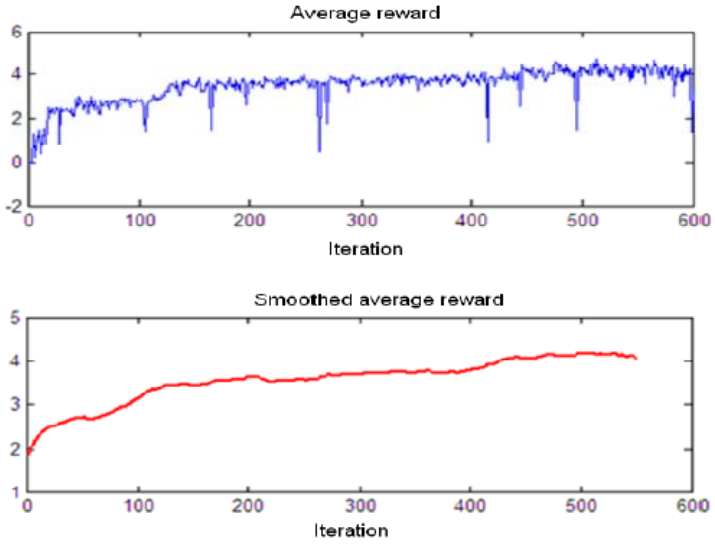


Figure 4.4 Learning curves associated with the driving environment c in Figure 4.3. An ε-greedy algorithm was used for action selection. Top row shows the average reward in each learning episode and the bottom row shows the average of the top row in a 50 iterations window. Parameters of the Q-value updating formula were α= 0.3 and γ=0.9.

and neuroscience. From AI point of view, many previous works (like RL, planning, probabilistic frameworks, etc) has tried to propose methods for performing complex tasks. "How complex tasks could be learned?" has been the question of the past research in AI. Now we state this question as "How complex tasks could be learned when an agent has bounded rationality because of bounded sensory power and how does it affect its behavior?".

Our proposed solution is in abstract form. Next step will be to apply it to visual complex tasks where agent has to perform a richer set of physical actions in natural or synthetic visual interactive environments. Another extension is to implement the units of the model presented at Figure 4.1 like GIST extractor. Fortunately recently in (Oliva & Torallba, 2006) a fast approach has been proposed for that purpose. It also remains to define and implement higher cognitive processes in higher visual processing unit of the model.

Chapter 5
Cost-sensitive Learning of Top-down Modulation for Attention Control

Ali Borji, Majid N. Ahmadabadi, Babak N. Araabi

Published in:
Journal of Machine Vision and Applications, In press
ECCV2008 Workshop on Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environments

Abstract. A biologically-inspired model of visual attention known as basic saliency model is biased for object detection. It is possible to make this model faster by inhibiting computation of features or scales which are less important for detection of an object. To this end, we revise this model by implementing a new scale-wise surround inhibition. Each feature channel and scale is associated with a weight and a processing cost. Then a global optimization algorithm is used to find a weight vector with maximum detection rate and minimum processing cost. This allows achieving maximum object detection rate for real time tasks when maximum processing time is limited. A heuristic is also proposed for learning top-down spatial attention control to further limit the saliency computation. Comparing over five objects, our approach has 85.4% and 92.2% average detection rates with and without cost respectively which are above 80% of the basic saliency model. Our approach has 33.3 average processing cost compared with 52 processing cost of the basic model. We achieved lower average hit numbers compared with NVT but slightly higher than VOCUS attentional systems.

Keywords Selective perception, Top-down attention, Bottom-up attention, Basic saliency model, Object detection

5.1 Introduction

Both machine vision and biological vision systems are faced with the problem of processing enormous amount of visual information they receive at any given time. Attentional selection provides an efficient solution to this problem by proposing a small set of scene regions worthy

further analysis to higher-level and cognitive processes like scene interpretation, object recognition, decision making, etc.

From a large body of literature in neurosciences and psychology, it is now known that attention is the process of selecting and gating visual information based on the saliency in the image itself (bottom-up) and on the prior knowledge about the scene (top-down). While bottom-up visual attention is solely determined by the basic and low-level physical characteristics of a scene- like luminance contrast, color contrast, orientation and motion- top-down attention on the other hand is influenced by the task demands, emotions, expectations, etc. Bottom-up component of the visual attention is mainly examined by the early visual areas of the brain like LGN, V1 and V2 (Desimone & Duncan, 1995, Li, 2002). Top-down attentional signals are largely derived from a network of parietal and frontal areas like frontal eye field (FEF), supplementary eye field (SEF) and lateral parietal cortex (Corbetta & Shulman, 2002). In daily life, these two mechanisms interact together to direct our attentional behaviors. Rather than acting in spatial domain (Posner, 1980, Posner & Cohen, 1984), visual attention could also be directed to particular features (Maunsell & Treue, 2006) and objects (Kanwisher & Driver, 1992).

As in biology, solutions in machine vision and robotics are limited in terms of processing huge amount of visual sensory information which is very time consuming. That is mainly because of the serial processing mechanisms used in the design of such creatures. This limitation necessitates engineering attentional control mechanisms in the brain of agents especially when they are supposed to act in real-world situations, which means to guarantee a limited response time. Some of the applications of attention in computer vision and robotics are object recognition, image compression, image matching, image segmentation, object tracking, active vision and human-robot interaction in addition to robot navigation and localization.

So far most experimental studies have addressed understanding bottom-up mechanisms of visual attention. That is probably because these mechanisms are mainly objective. On the other hand, top-down mechanisms show subject-to-subject variability which makes them difficult to tackle (Yarbus, 1967). As a result, computational studies have been concentrated more on modeling bottom-up mechanisms due to lack of abstract knowledge on top-down component of visual attention. In our work, we propose an approach for learning top-down attentional modulations without explicit mathematical formulation of impact of an object or its surrounding in saliency.

From a behavioral perspective, some of the human behaviors are learned by exposing them to a set of offline data. For example for learning to drive, a person familiarizes himself with traffic signs and their associated meanings and then uses this knowledge when doing real-time driving. Merging these two learning phases- offline learning of rules and signs and learning to apply them interactively and online- seems to be the best approach for doing this complicated task. According to this logic, our attention system could be considered as a basic component of a larger system which provides top-down signals when requested by a higher cognitive layer.

We consider attention control as an optimization problem in which an artificial agent has to devote its limited processing power to the most relevant items or dimensions of the sensory space. The basic idea is that an agent is looking for an object of its interest in the scene and therefore has to bias its bottom-up attention system in order to detect that object *efficiently* and *fast*. We follow a data-driven approach for learning top-down attention, where top-down attentional signals are learned from a set of training images containing an object in clutter. This knowledge can later be

used when an agent is doing a task online and needs to attend to different objects in different situations. The result of optimization will be ignoring some features (sensors) in addition to finding their relative importance in construction of a final saliency map.

Our approach is an extension of the basic saliency model (Itti et al., 1998), which is based on the idea of saliency map, an explicit two-dimensional topographical map that encodes stimulus conspicuity or saliency at every scene location. Saliency model is purely data-driven and simply selects some spatial locations without using any feedback mechanism or top-down gains. In particular, we introduce two contributions. First, the basic saliency model is revised in a way which allows selection and weighting of different scales. Then an evolutionary algorithm is used for biasing this model toward specific objects in cluttered scenes while considering the costs of operations of the saliency model. This allows an agent to use its certain limited processing resources efficiently in terms of achieving maximum object detection rate. In our second contribution, we propose a method to reduce saliency computation and therefore faster object search in natural scenes. It is based on the observation that, in some tasks objects appear in specific spatial locations with respect to the observer. For example in driving, traffic signs mostly appear in the right visual field of the driver. In addition to these, performances of our approach, basic saliency model as well as a general benchmarking approach for object detection are compared over disrupted images with several types of noises.

The rest of this chapter is organized as follows. In Sect. 2 related works for learning top-down attention control and traffic sign detection are reviewed. Basics of our method for learning top-down feature-based and spatial attention control are presented in Sect. 3. Experiments and results are shown in Sect. 4. Sect. 5 discusses the results and finally, Sect. 6 summarizes and concludes the paper.

5.2 Related works

A broad range of modeling works is reported in the literature of visual attention, however little research is reported on learning and development of attention control. In this section we review some studies which are directly related to ours, mainly those focused on learning aspects of visual attention control. In order to situate our work among previous works on traffic sign detection, some successful studies from this domain are also reviewed.

5.2.1 Learning top-down attention control

Evidence toward the idea that attention could be learned by biological mechanisms is proposed in (Nakayama et al., 2004). In a behavioral task, authors showed that a short memory system is responsible for rapid deployment of visual attention. They suggest that fast and transient component of visual attention is flexible and capable of learning simple relationships (Nakayama, 1989) and is dependent on the previous experiences of the subjects (Kristjansson & Nakayama, 2003). Human subjects were supposed to answer a question about a feature of a specific visual item in a synthetic search array. Subjects had lower reaction times when this feature remained the same throughout successive trials (Maljkovic & Nakayama, 1994).

In (Rybak et al., 1998), it is postulated that human eyes move and successively fixate at the most salient parts of an image during visual perception and recognition. These parts are then processed with the highest resolution. Simultaneously, the mechanism of visual attention chooses the next eye position using information extracted from the retinal periphery (Klein, 2000). Eye

movement behavior has been shown to be task-specific and context-dependent in humans (Yarbus, 1967). In a behavioral task, human subjects were asked a question about a scene presented to them and their saccadic eye movements were recorded. Depending on the question, subjects had different eye movement patterns. For example when asking to judge about age of persons in a scene, eye movements were mainly focused on faces. This experiment points toward another evidence that top-down attention can be learned.

One of the outstanding works on visual attention, known as the basic saliency model, is proposed by Itti et al. (Itti et al., 1998) which is an extension and implementation of an earlier model of visual attention by Koch and Ullman (Koch & Ullman, 1985). Simplicity and little processing time are two main advantages of this model. It has been used to explain behavioral data on simple synthetic and static search arrays and also dynamic natural stimuli like movies and games (Peters & Itti, 2008). To add top-down capabilities to this model, a task-driven model of visual attention is proposed in (Navalpakkam & Itti, 2005)**0**. Given a task definition in the form of keywords, this model first determines and stores task-relevant entities in working memory using prior knowledge stored in a long-term memory. It then attempts to detect the most relevant entity by biasing its visual attention system with the entity's learned low-level features. It attends to the most salient location in the scene and attempts to recognize the attended object through hierarchical matching against object representations stored in the long-term memory. It updates its working memory with the task relevance of the recognized entity and updates a topographic task relevance map with the location and relevance of that entity. In this study, we aim to build our top-down attention system upon the basic saliency model and bias it for detection of different objects in natural scenes.

In (Frintrop, 2006), Frintrop et al. have introduced a new computational attention system known as VOCUS for efficient and robust detection of regions of interest in images. In their approach, the selection of an image region is determined by two kinds of influences: bottom-up and top-down cues. Bottom-up cues are determined by local contrasts and by uniqueness of a feature. Top-down cues depend on the features of a pre-specified target. Bottom-up and top-down down saliency maps are then linearly added and weighted to form a final saliency map. They have shown that VOCUS is robust and applicable to real-world scenes.

Basic saliency model concentrates on computing bottom-up attention. Recently, Navalpakkam and Itti (Navalpakkam & Itti, 2006) have introduced a newer version of the basic model by adding top-down capabilities to it. The idea is to learn feature values of a target object from a training image in which the target is indicated by a binary mask. By considering the target region as well as a region in its close surrounding, their system learns feature values from different feature maps. During object detection, this feature vector is used to bias the feature maps by multiplying each map with the corresponding weight. Thereby, exciting and inhibiting as well as bottom-up and top-down cues are mixed and directly fused into the resulting saliency map.

In contrast to the above mentioned two techniques (Frintrop, 2006, Navalpakkam & Itti, 2006), instead of only finding the appropriate weights, we also incorporate processing costs of the feature channels to force an optimization process to choose the feature vectors with high detection rate and low cost. Basic saliency model does not allow scale selection because center-surround inhibition in this model is implemented by subtraction of scales from each other. We revise the basic model by implementing surround inhibition in each scale independent of others. This not only allows scale weighing but also allows selection among scales for saliency detection.

Inhibition of those scales which are not important in detection of an object results in faster object detection and less computation. Furthermore associating costs to operations of the basic saliency model allows optimal use of computational resources of the agent. For example when an agent has a certain amount of computation power, it must efficiently select/weight features and scales of the saliency model in order to achieve the maximum detection rate. Our approach in this paper provides such a capability.

5.2.2 Traffic sign detection and recognition

Traffic sign recognition (TSR) can be considered as part of the bigger problem of autonomous driving. An autonomous driver system relies on vision-based recognition of surrounding area in order to make driving system function as the feedback provider for control of steering wheel, accelerator, brake, etc. Besides the application of TSR in autonomous vehicles, it can also serve as an assistant driver to notify the driver about approaching a traffic sign or his risky behavior (like driving above speed limit). Driving is the best example of a complex task which involves many cognitive behaviors and most importantly attention. Due to this, we consider visual attention as a part of a TSR system for fast detection of traffic signs.

Many researchers have developed various techniques for automatic road traffic sign recognition. Regarding the detection problem, different approaches have been proposed. A few approaches rely solely on gray-scale data. Gavrila (Gavrila, 1999) employs a template-based approach in combination with a distance transform. Barnes and Zelinsky (Barnes & Zelinsky, 2004) utilize a measure of *radial symmetry* and apply it as a pre-segmentation within their framework. Since radial symmetry corresponds to a simplified (i.e., fast) circular Hough transform, it is particularly applicable for detecting possible occurrences of circular signs. The majority of recently published sign detection approaches make use of color information (de la Escalera et al., 2003, Paclik et al., 2000, Gomez & Fuentes, 2007). They share a common two-step strategy. First, a pre-segmentation is employed by a thresholding operation on a color representation. Some authors perform this directly in RGB space, others apply its linear or nonlinear transformations. Subsequently, a final detection decision is obtained from shape based features like corners or edge features, applied only to the pre-segmented regions. The most discriminating colors for traffic signs include red, orange, yellow, green, blue, violet, brown and achromatic (Gomez & Fuentes, 2007). A joint treatment of color and shape has been proposed by Fang et al. (Fang et al., 2003). The authors compute a feature map of the entire image frame, based on color and gradient information, while incorporating a geometry model of signs. Detection of traffic signs in only a single image has three problems (de la Escalera & Moreno, 1997): (1) information about positions and size of traffic signs, which is useful to reduce the computation time, is not available (2) it is difficult to detect a traffic sign correctly when temporary occlusion occurs and (3) correctness of the detection is hard to verify. In order to handle these issues tracking techniques have been developed for traffic sign detection. In (de la Escalera & Moreno, 1997), image sequences are utilized for recognition of traffic signs.

Discriminating shape and colors of traffic signs make them easily recognizable by humans. Same factors bring the idea of applying the basic saliency model for detection of traffic signs which we follow in this work. Not only we analyze the capability of the saliency model for detecting traffic signs, but also we evaluate its efficiency for natural object detection.

The main reason why we propose the basic saliency model of visual attention as a subsystem of a typical TSR system for detection of traffic signs is because of its fast computation time. This does not necessarily mean that it could outperform above mentioned methods for detection of traffic signs. There are two reasons for this. First, basic saliency model is based on abstract biological findings from human vision and is designed in a way to be fast like human visual attention. It is a general purpose system for detection of salient image regions. Second, basic saliency model in its current form does not consider structure and shape of objects. This weakness makes it inferior (bus faster) to single purpose solutions for detection of specific objects like faces or traffic signs. Adding learning capabilities to the basic saliency model for enabling it to also consider shape of different objects demands more research and goes beyond the scope of this paper.

5.3 Learning top-down attentional modulations

In this section we present our goal-directed visual attention system. Top-down influences play an important role in human visual attention rather than bottom-up cues. Top-down sources of visual attention are mainly determined by the experiences, motivations, emotions, expectations and goals. For example a hungry person is focused on foods.

Saliency model without biasing selects a spatial location in a feed-forward manner. Saliencies in feature dimensions- intensity, orientation and color- are computed independently and are fused into a single final saliency map. The top-down extension in our model includes a search phase to learn feature weights to determine which features to enhance and which ones to inhibit. The weighted features contribute to modulate the bottom-up saliency map highlighting regions with target-relevant features. Instead of only finding the appropriate weights, we also incorporate processing costs of the feature channels to force the optimization process to choose the feature vectors with high detection rate and low cost. When an agent has a certain amount of processing resources it could bypass computation of irrelevant feature channels or scales of the basic saliency model. This is actually a constrained optimization problem which we solve by global optimization techniques.

5.3.1 Revised saliency-based model of visual attention

To furnish the basic saliency model for our purpose, i.e biasing it for object detection, a Gaussian pyramid (Burt & Adelson, 1983) with 6 scales (s_0 .. s_5) is built in each feature channel of the image by successively filtering the image with a Gaussian low-pass filter (GLF) that is then subsampled, i.e s_{i+1} has half the width and height of s_i. Three features are considered; intensity, orientation and color. A surround inhibition operation is then applied to all scales of each pyramid to enhance those parts of the image which are different from their surroundings. To implement surround inhibition, basic saliency model (Itti et al., 1998) subtracts coarser scales from finer ones. Since we would like to select scales rather than weighting them, we dissociate scales and then apply the surround inhibition over each scale separately. This is not possible in the basic model because scales are dependent to each other and hence cannot be inhibited or selected. Surround modulated images in each scale of a pyramid are upsampled to a predetermined scale (here the largest scale, s_0) and are then added. To accomplish surround inhibition (SI), we designed a nonlinear filter which compares the similarity of each pixel with the average of its surrounding window and then inhibits the center pixel as:

$$Im' = SI(Im), \tag{5.1}$$
$$Im'(x,y) = \max\left(0, Im(x,y) - mean\left(surround\left(Im(x,y)\right)\right)\right), \forall\, x,y \in Im$$

where surround is a spatial n×n mask around pixel *Im(x,y)*. *Im'(x,y)* is the new value of the pixel. Fig. 1 demonstrates application of the surround inhibition operation on some test images. Here, all scales of the intensity pyramid are surround inhibited and then added to form the saliency map at the intensity channel. As Fig. 1 shows, this operation has resulted in marking the salient areas with respect to the rest of each image.

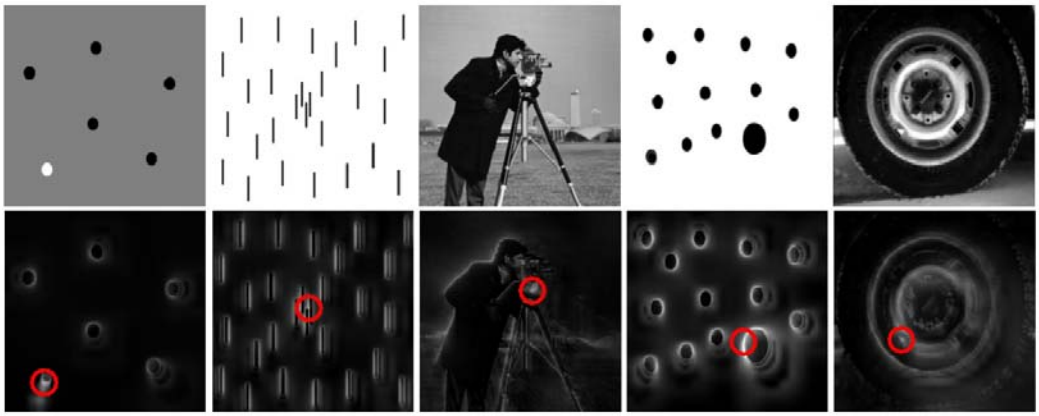


Figure 5.1 Surround inhibition operation using (1). Top row shows five test images and bottom row shows the detected salient regions with surround window sizes (n) from left to right as 7, 7, 3, 7 and 5. Red circles illustrate the most salient locations.

We used a MATLAB® implementation of the basic saliency model and revised it for our purpose[1]. The whole revised saliency model is shown in Fig. 5.2.

[1] Saliency toolbox homepage. http://www.saliencytoolbox.net/

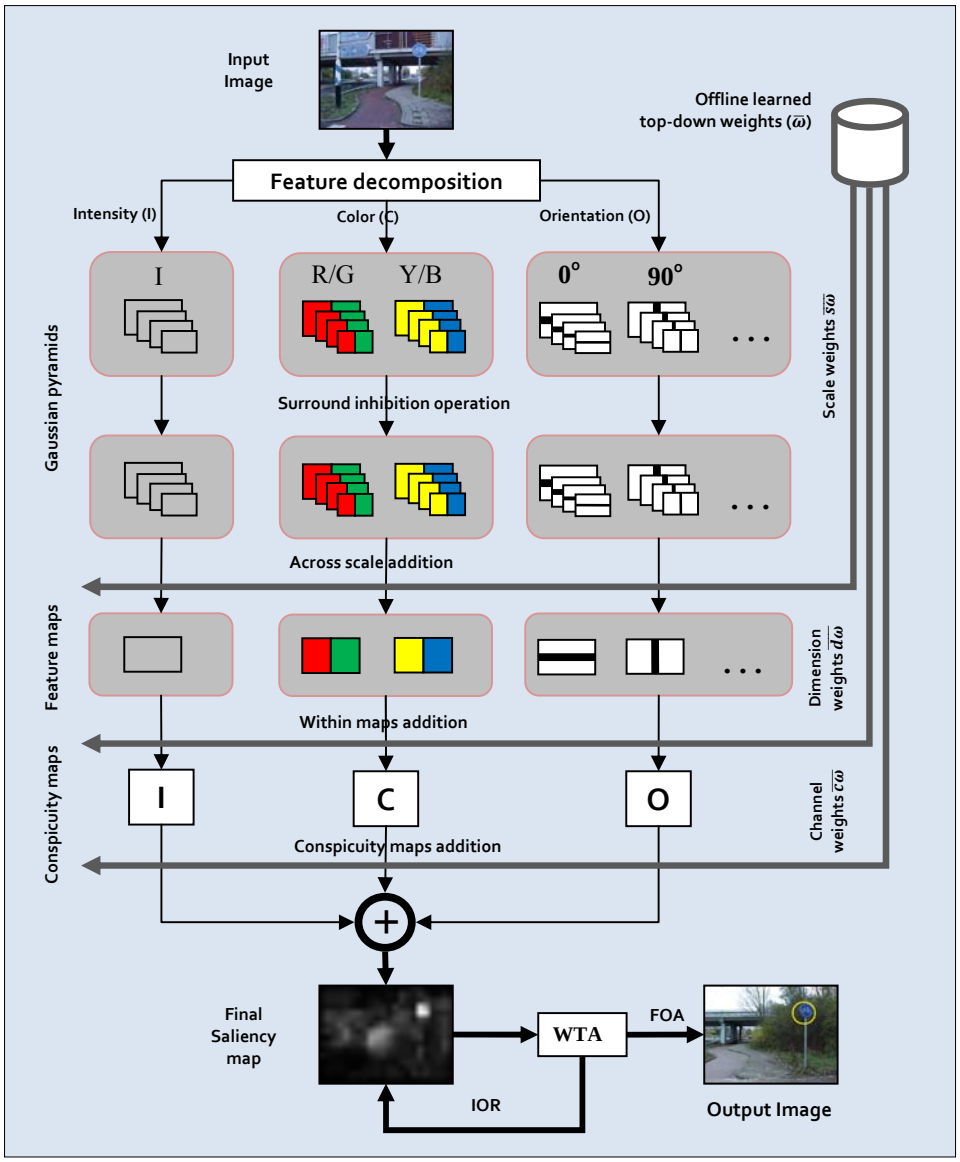


Figure 5.2 Proposed top-down saliency model. Input image is decomposed to several feature channels and dimensions within each feature channel. Here, 1 dimension for intensity channel, 2 red/green and yellow/blue dimensions for color channel and 4 dimensions for orientation channel are used. Image pyramids are built in each feature dimension by successively low-pass filtering and subsampling the image. Surround inhibition operation (1) is applied to all scales of a channel. In the next step, maps across scales are weighted and summed to form feature maps (2,3). Then, feature maps are normalized, weighted and summed again to form conspicuity maps (4). Final saliency map is derived by normalizing, weighting and summing these conspicuity maps (5). Maximums at the final map determine the focus of attention (FOA). After a time constant, the most salient location is inhibited to allow other locations grab attention (IOR). Contributions of scales in dimensions, dimensions in feature channels and feature channels in the final saliency map are determined by a learned weight vector ($\bar{\omega}$).

The input image to the system is decomposed into 3 feature channels: Intensity (*I*), Color (*C*) and Orientation (*O*). Color channels are calculated as follows. If *r*, *g* and *b* are the red, green and blue dimensions in RGB color space, then $I = (r + g + b)/3$, $R = r - (g + b)/2$, $G = g - (r + b)/2$, $B = b - (r + g)/2$, and $Y = r + g - 2(|r - g| + b)$ (negative values are set to zero). Local orientations (O_θ) are obtained by applying Gabor filters to the images in the intensity pyramid *I*. These operations are shown in (2). P_s is the feature at scale s. P could be intensity (I), Red (R), Green (G), Blue (B), Yellow (Y) or orientation (O). $O_{\theta,s}$ is the orientation channel at orientation θ and scale s.

$$\begin{aligned} F_{I,s} &= SI(I_s) \\ F_{RG,S} &= SI(R_s - G_s) \\ F_{BY,s} &= SI(B_s - Y_s) \\ F_{\theta,s} &= SI(O_{\theta,s}) \end{aligned} \tag{5.2}$$

In (2), $F_{I,s}$, $F_{RG,s}$, $F_{BY,s}$ and $F_{\theta,s}$ are the intensity, red/green, yellow/blue and orientation channels in scale *s*, respectively. SI is the surround inhabitation operation in (1). These feature maps are summed over and sums are normalized again:

$$F_l = N(\sum_s (s\omega)_s . F_{l,s}) \text{ with } l \in L_I \cup L_C \cup L_O \tag{5.3}$$

$$\text{and } L_I = \{I\}, L_C = \{RG, BY\}, L_O = \{0^0, 45^0, 90^0, 135^0\}$$

where $(s\omega)_s$ is the weight of scale *s*. $N(.)$ is an iterative, nonlinear normalization operator, simulating local competitions between neighboring salient locations **0**. In each feature channel, feature dimensions contribute to the conspicuity maps by weighting and normalizing once again (4).

$$C_p = N\left(\sum_{l \in L_p} (d\omega)_p . F_l\right), \qquad p \in \{I, C, O\} \tag{5.4}$$

Variable $(d\omega)_p$ in (4) determines weight of a dimension within feature channel *p*. All conspicuity maps are weighted and combined at this stage into a final saliency map (5).

$$SM = \sum_k (c\omega)_k . C_k , \qquad k \in \{I, C, O\} \tag{5.5}$$

$(c\omega)_k$ in (5) weights the influences of feature channels in the final saliency map. The locations in the saliency map compete for the highest saliency value by means of a *Winner-Take-All* (WTA) network of integrate and fire neurons (Itti & Koch, 2001). The winning location of this process is attended to and the saliency map is inhibited at this location. Continuing WTA competition, next most salient location is attended to and so on to form a scanpath of successive overt attentions.

5.3.2 Offline learning of top-down attentional modulations

Real-world systems have limited amount of computational resources due to serial processing mechanisms used in their designs. Therefore it is necessary to optimize use of these resources. For the agent to make optimum use of its processing resources, in addition to a weight, a cost is also associated to each feature channel and image resolution. Weight and cost vectors are represented by $\bar{\omega}$ and $\bar{C}$, respectively in (6). Weight vector determines weights of feature channels, dimensions within channels and also scales in image pyramids respectively. Elements in the cost vector correspond to the associated feature or scale determined by the weight vector.

$$\begin{aligned} \bar{\omega} &= \left(\overline{c\omega}, \overline{d\omega}, \overline{s\omega}\right), |\bar{\omega}| = 16, |\overline{c\omega}| = 3, \left|\overline{d\omega}\right| = 7, |\overline{s\omega}| = 6 \\ \overline{d\omega} &= \left(\overline{d\omega I}, \overline{d\omega C}, \overline{d\omega O}\right), \left|\overline{d\omega I}\right| = 1, \left|\overline{d\omega C}\right| = 2, \left|\overline{d\omega O}\right| = 4 \\ \bar{C} &= (c_1, c_2, \dots, c_{16}) \end{aligned} \tag{5.6}$$

In (6), $\overline{c\omega}$, $\overline{d\omega}$, $\overline{s\omega}$ are weight vectors for feature channels, dimensions within channels and scales, respectively. $\overline{d\omega I}, \overline{d\omega C}, \overline{d\omega O}$ are weight vectors for intensity, color (red/green and yellow/blue) and orientation (0^o, 45^o, 90^o and 135^o) dimensions. Our aim is to find a vector of weights ($\bar{\omega}$) determining the contributions of features and image resolutions to detect a specific object in a set of images while taking into account their costs ($\bar{C}$). Thus, the optimum weight vector must satisfy these two objectives (1) It must enable the saliency model to detect an object of interest correctly (maximum detection rate) and (2) Its associated feature vector, must have the least computation (minimum processing cost). For this purpose, we follow a data-driven approach. First, optimal weight vector is sought to satisfy the above two objectives over a training image dataset and is then evaluated over a separate test set of images. Assume that training set contains *M* images with an object of interest tagged in them as:

$$T = \{Im_1, Im_2, \dots, Im_M\} \tag{5.7}$$

An example fitness function which satisfies the above mentioned two objectives is shown in (8).

$$\begin{aligned} f(\bar{\omega}) &= \frac{1}{M}\left(\sum_{i=1}^{M} \text{norm}\left(\text{Saliency}\left(Im_i, r(\bar{\omega})\right) - t_i\right)\right)\left(u(\bar{\omega}) \cdot \bar{C}\right) \\ r(\omega_i) &= \begin{cases} \omega_i, & \omega_i > \alpha \\ 0, & \text{otherwise} \end{cases}, \quad u(\omega_i) = \begin{cases} 1, & \omega_i > \alpha \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{5.8}$$

In (8), *M* is the size of the train set, norm(.) is the Euclidean distance between two points in an image. Saliency is a function which takes an image and a vector of weights as input and determines the most salient location in the image. t_i is the location of the target object in the *i*-th image. *u(.)* is the step function and is 1 when a feature channel or resolution is used. *r(.)* is the ramp function and zeros its input when it is smaller than a threshold. Operator $\cdot$ is the dot product. Since we are going to compare the saliency detection with and without costs, results are also reported using the fitness function of (9) when costs are ignored and the single objective is to maximize the detection rate.

$$f(\bar{\omega}) = \frac{1}{M}\left(\sum_{i=1}^{M} \text{norm}(\text{Saliency}(Im_i, r(\bar{\omega})) - t_i)\right) \tag{5.9}$$

We also consider a case when the agent has a limited computational power and has to select those feature channels or scales of the basic saliency model when the accumulated cost does not exceed a cost limit (Q). This is a constrained optimization problem in which goal is to maximize detection rate but with a constraint which is the cost limit:

$$\begin{aligned} &f(\bar{\omega}) = \tfrac{1}{M}\left(\textstyle\sum_{i=1}^{M} \text{norm}\left(\text{Saliency}\left(Im_i, r(\bar{\omega})\right) - t_i\right)\right) \\ &u(\bar{\omega}) \cdot \bar{C} \ \leq Q \end{aligned} \tag{5.10}$$

Note that, a lower fitness value for the above functions means that it has better performance. For minimizing fitness functions, an algorithm known as comprehensive learning particle swarm optimization (CLPSO (Liang et al., 2006)) is used. CLPSO is simple, easy to implement and has been applied to a wide range of optimization problems. Its fast convergence and better performance over multi modal functions are the reasons why we chose this optimization algorithm. First, particles of CLPSO (each particle is a 16D weight vector) are randomly initialized in the range [0 6]. Salient locations are detected by applying each particle to the saliency model to calculate its fitness. Then, through CLPSO operations, particles are evolved in order to minimize the fitness values. In cost-limited case (10) when an individual violated the constraint condition, its fitness was assigned a very large positive number. Table 1 shows parameters of CLPSO used in the experiments.

Table 5.1 Parameters of CLPSO used in experiments

parameter	value
particles	300
max iterations	40
dimensions	16
$[\min(x_i)\ \ \max(x_i)]$	[0 6]
refreshing gap	5
$[\omega\ c\ p_c\ v_{max}]$	[0.9 1.5 0.3 5]

5.4 Experiments and results

Humans have the capability to attend to features or dimensions of perceptual space or could select the spatial locations which usually contain objects **0**. The same capability is highly desired for artificial systems. In this section by means of 4 experiments we show how our algorithm can be used for both feature-based and spatial attention control.

5.4.1 Learning top-down feature based attention control

Three experiments demonstrate the strength of our algorithm for feature-based attention control. First 2 experiments are pop-out and conjunction search tasks (Wolfe, 1998) and object detection in natural scenes. Search performance (percentage of detection rate), average hit number and cost of our method are compared with the basic saliency model in both without-cost (9) and with-cost (8) cases. In the third experiment, performance of our method is compared with the basic saliency model as well as Template Matching (TM) approach over noisy images.

In with-cost case, feature costs were defined based on relative computational costs of feature channels and image resolutions as $\bar{C} = [3\ 1\ 4, 3\ 3, 1, 4\ 4\ 4\ 4, 6\ 5\ 4\ 3\ 2\ 1]$. For example, a color channel needs more computation than an intensity channel but less compared to an orientation channel. Or computation of surround inhibition in scale s_0 is more expensive than those of other scales in the Gaussian pyramid.

Experiment I. Pop-out and conjunction search tasks In a pop-out or conjunction search task, a search array is presented to the system. It has to determine which item in the search array is different from other items. Then its reaction time is measured. Items in the search array differ in one dimension only in a pop-out task, while in a conjunction search task, items differ in more than one dimension which makes the task harder. Psychological data have shown that *reaction times (RT's)* of human subjects remain constant with increasing the number of items in the search array. In contrast, in a conjunction search task, reaction times increase linearly with size of the search array.

Revised saliency model was trained using CLPSO for known saliency detection over 2 pop-out and 3 conjunction search arrays shown in Fig. 3. Fig. 4 shows biasing weights over the synthetic search arrays in both without-cost and with-cost cases after CLPSO convergence. CLPSO is trained 5 times and averages are shown. In this experiment, size of the surround inhibition window was 5. In all search arrays target was successfully detected in first fixation.

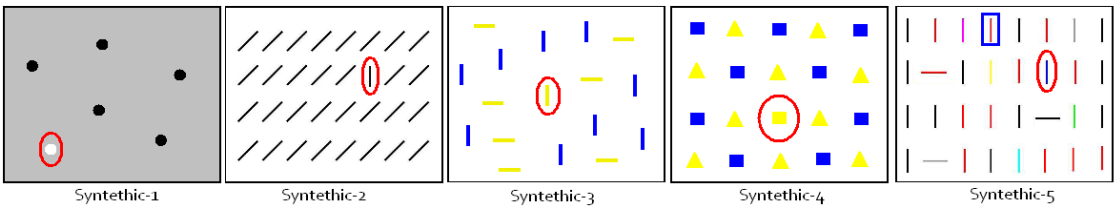


Figure 5.3 Synthetic search arrays used in experiment I. Two left arrays are pop-out and 3 right ones are conjunction search tasks. Target item is shown with red ellipse. Blue rectangle in the fifth array is the first saccade of the basic saliency model.

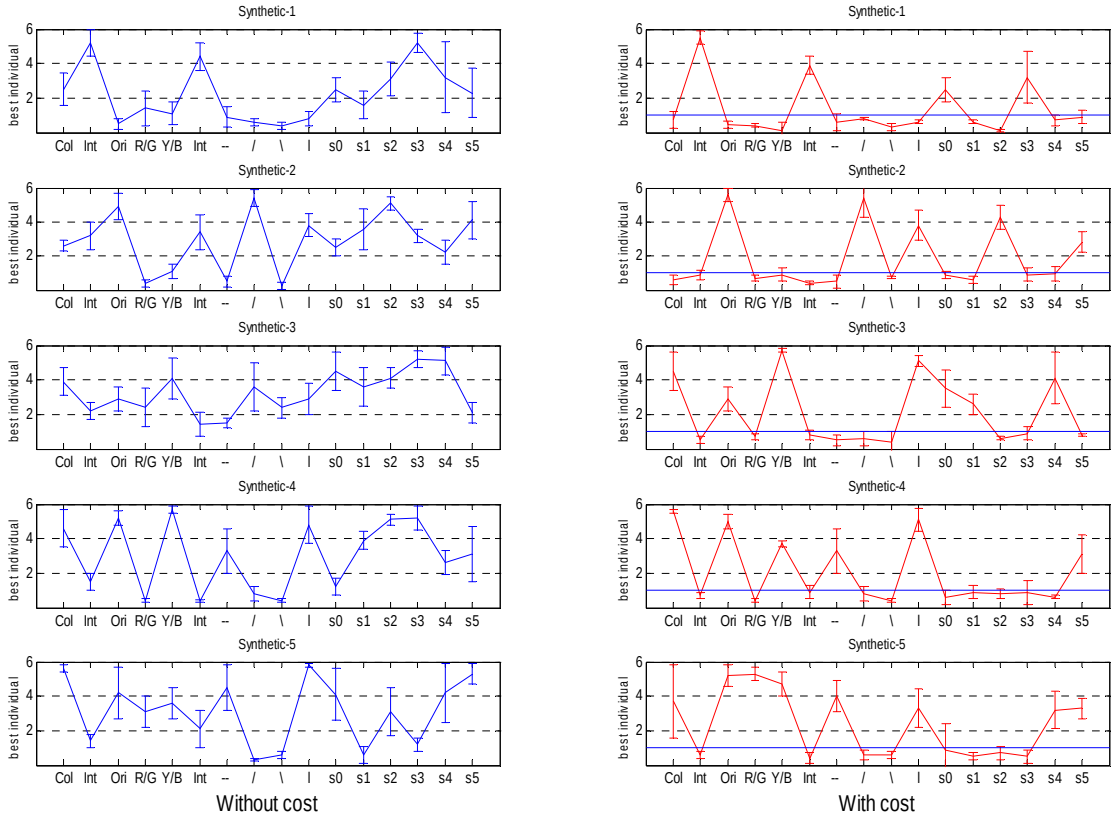


Figure 5.4 Learned weights after CLPSO convergence over synthetic search arrays. Left column shows the weights learned in without-cost case and right column shows with-cost case. Blue horizontal lines in the right column show lines α=1 (threshold parameter in (8)). Employed features are: Color, Intensity, Orientation, Red/Green and Yellow/Blue dimensions within color channel, a single intensity dimension, 4 orientation dimensions within orientation channel and 6 scales.

As Fig. 4 shows, in the first search array with white dot among black dots, intensity channel has the highest weight in both cases. In with-cost case of this array, only intensity channel and scales s_0 and s_3 are selected resulting in mean total cost of 11. Since orientation is not important in saliency of the target it is ignored by CLPSO. In the second array with vertical bar among oblique bars, in without-cost case, orientation channel and two dimensions (45^o and 90^o) have the highest weights. In with-cost case other channels except orientation are suppressed to reduce cost from 41 to 17. Intensity channel is ignored because intensities of target and distracters are the same. Since

the target in the third array is the yellow bar, yellow/blue color dimension have the highest weight. For this array scales s_0 to s_4 are more important. In the forth array, color channel and yellow/blue dimension have the highest weights. In with-cost of this array only yellow/blue, 0^o and 90^o dimensions are selected. While in the first two pop-out search arrays, the basic saliency model can also detect the target, in the conjunction search arrays, CLPSO was trained to selectively attend to a target which is not necessarily the attended item by the basic saliency model. For example, while in the last synthetic image, the basic saliency model selects the bar in the rectangle, we purposefully chose the blue bar to become salient. Since there are several colors available in this array, both dimensions within the color channel have got high weights. Weights for intensity channel and orientations (45^o and 135^o) are very low. In with-cost case of this array, intensity, orientation and scales (s_0 to s_3) which are not discriminating the target object are not selected by the evolutionary process.

Experiment II. Natural object and traffic sign detection Proposed method was also evaluated for natural object and traffic sign detection on cluttered scenes[2]. We used 3 traffic signs (bike, crossing and pedestrian) and 2 objects (coke and triangle). Number of images for bike, crossing, pedestrian, triangle and coke were 70, 45, 55, 69 and 42, respectively. Sizes of images were 360×270 pixels. Fig. 5 illustrates sample signs and objects in natural scenes.

Figure 5.5 Sample objects used in experiment II. From left to right: bike, crossing, pedestrian, coke and triangle. Target is shown by the yellow circle.

CLPSO was trained over 10 random images for each object and then the best weight vector was tested over the remaining images of that object. Results are reported over 5 runs with random train images. Fig. 6 shows fitness values for both without-cost and with-cost cases during CLPSO convergence over training sets. Derived weight vectors in both cases are shown in Fig. 7. Window size of surround inhibition was 7.

[2] Images for this experiment were selected from databases available at http://ilab.usc.edu/imgdbs and http://www.cs.rug.nl/~imaging/databases/traffic_sign_database/traffic_sign_database.html.

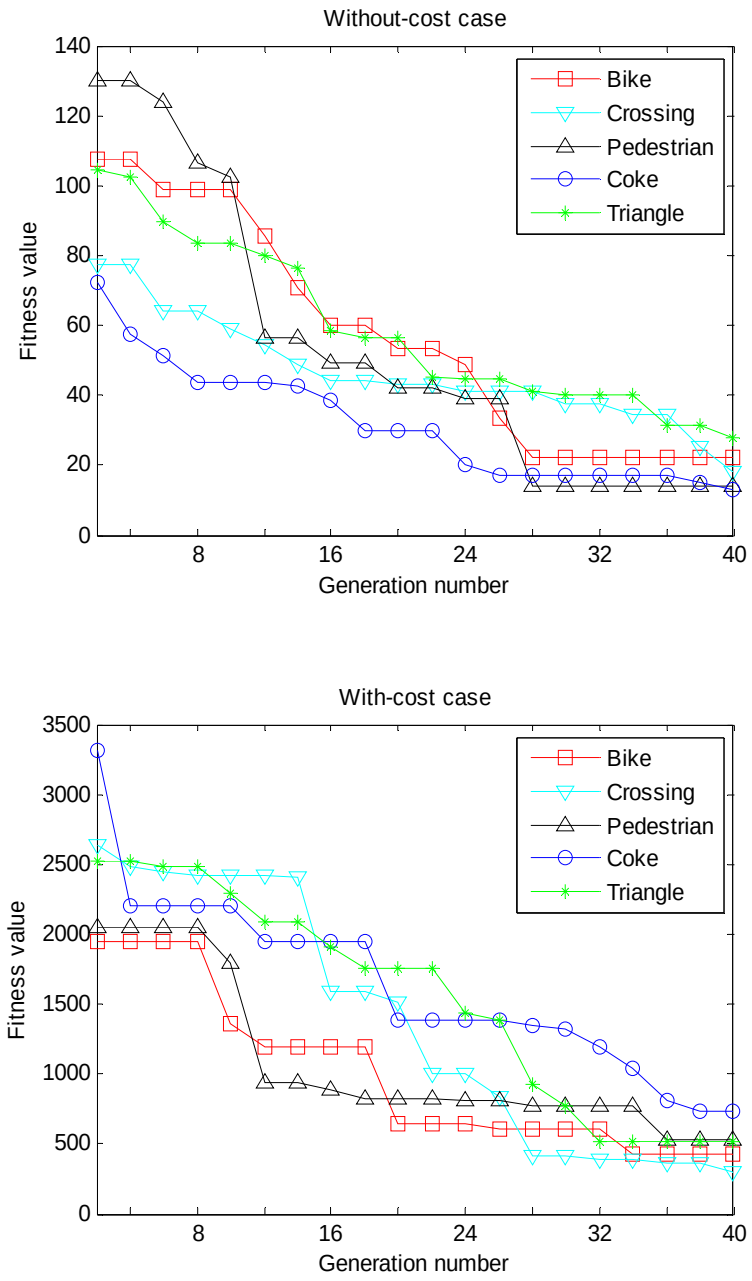


Figure 5.6 CLPSO convergence for traffic signs and natural objects in experiment II in both without-cost (top) and with-cost (bottom) cases.

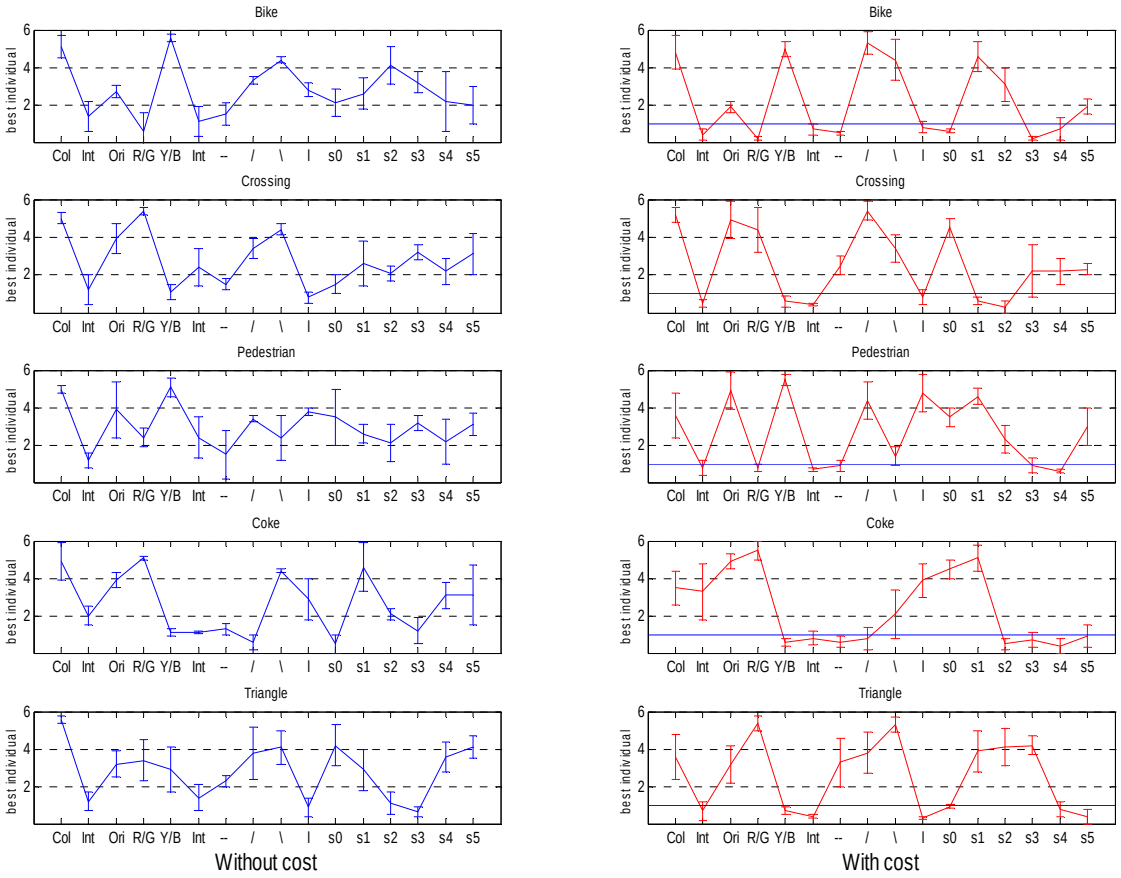


Figure 5.7 Learned weights after CLPSO convergence over traffic signs and objects. Left column shows the weights learned in without-cost case and right column in with-cost case. Blue horizontal line in right column shows the line α=1 (threshold parameter in (8)). Values below this line are considered zero to evaluate the detection rates.

For bike, in without-cost case, color (yellow/blue) and orientation (45° and 135°) channels have the highest weights. Middle scales (s_1 to s_3) are more informative for detection of this object. In with-cost case, the optimization process has decreased the contribution of intensity and horizontal orientation. For crossing, again color channel is selected in without-cost case. For this sign, red/green channel has higher weight. Orientation dimensions (0°, 45° and 135°) which appear in shape of the crossing sign have higher weights. Since almost in all images of this sign, triangle is toward up, these orientations are stable features and have got higher weights than other orientation dimensions. In with-cost case of this sign, color and orientation channels have survived in the evolutionary process. In this case, the largest scale s_0 has the highest weight. For pedestrian sign, color (yellow/blue) and orientation channels are selected in both cases. Important orientations for this sign are 45° and 90°.

For the coke object, color (red/green) channel, orientation 135° and scale s_1 have higher weights in without-cost case. In both cases orientations 90° and 135° are important for detection of this object. For triangle object as in crossing sign in both cases, 0°, 45° and 135° orientations have

higher values again showing importance of these features in discrimination of triangular objects. Among color channels, red/green dimension has the highest weight.

Tables 2 and 3 show the average values of detection rates and average hit numbers using fitness functions in (8) and (9). An object was considered detected if a salient point was generated in a vicinity of 30 pixels around t_i. For detection of an object rather than the most salient point, 2 other locations generated by WTA were also considered. Hit number of an image Im_i for a known target t_i is the rank of the focus that hits the target in order of saliency. For example, if the second focus is on the target then its hit number is 2. Images with targets not detected in max fixations are not included in the averages. The average hit number for an image set is the mean of hit numbers of all images. A search method with lower average hit number is favored.

Table 5.2 Detection performances of biased saliency model in without-cost case and the basic saliency model over natural objects and traffic signs with max fixations equal to 3. Results are averaged over 5 runs with random training and test sets. Performance of the basic saliency model is over the test sets. Numbers in parentheses are standard deviations.

target	biased saliency model						basic saliency model	
	train			test			detection rate %	avg. hit number
	detection rate %	avg. hit number	fitness	# test images	detection rate %	avg. hit number		
bike	92.3(2.1)	1.4(0.4)	22.3(9.2)	60	90.2(2)	1.6(0.1)	81.8(0.6)	2.2(0.2)
crossing	96.7(1.2)	1.5(0.7)	18.1(5.4)	35	93.8(0.9)	1.5(0.2)	78.2(1.4)	2.5(0.3)
pedestrian	98.2(1)	1.2(0.2)	14.4(6.1)	45	94.2(1.1)	1.3(0.7)	83.3(1)	1.7(0.1)
coke	95.2(1.4)	1.3(0.3)	13.2(8)	59	92.2(2)	1.5(0.5)	80.9(0.4)	1.9(0.5)
triangle	92.5(2.3)	1.7(0.9)	27.8(11.4)	32	91(1.6)	1.8(0.4)	76.5(0.8)	2.2(0.2)

Table 5.3 Detection performance over natural objects and traffic signs in with-cost case with max fixations equal to 3.

target	train			test			
	detection rate %	avg. hit number	fitness	# test images	detection rate %	avg. hit number	computation cost
bike	87.8(1.2)	1.7(0.3)	421.9(25.6)	60	85.8(1.5)	1.8(0.2)	25.5(4.2)
crossing	84.2(1.9)	1.5(0.7)	302.1(47.1)	35	78.2(2.1)	2(0.3)	40.5(5.8)
pedestrian	91.6(1)	1.5(0.4)	531.9(54.3)	45	90.6(2)	1.7(0.4)	35.6(5.2)
coke	87.3(2.7)	1.8(0.6)	730(34.4)	59	87.1(1.1)	1.9(0.1)	32.2(7.3)
triangle	89.6(2.1)	1.4(0.2)	512.2(27.7)	32	85.6(1.2)	2.1(0.2)	32.7(6.1)

The basic bottom-up saliency model has always total cost of 52, since it uses all the channels and resolutions. On the other hand, biased model has the average cost of 33.3 over signs and objects, which is less than the cost of the basic saliency model. Average detection rate with 3 hits

of the basic saliency model is 80% while we achieved 92.2% without cost and 85.4% with cost. Results prove that biasing leads to higher recognition rate and lower cost than the basic saliency method. Average hit numbers of our approach are 1.54 (without-cost) and 1.9 (with-cost) while the basic saliency model has average hit number of 2.1. Performance of our model without biasing is nearly the same as the basic saliency model. Variance in the costs shows that cost is sensitive to train and test data.

Fig. 8 shows detection rate of cost-limited case (10) for several values of cost. As the cost limit increases, detection rate also increases. In this case the best individual learned with a certain amount of cost is applied to the test set. This figure helps an agent to assign its processing power to reach the accuracy level it needs. For example in order to reach 80% for bike detection, the agent should at least have 20 computational resources or costs. Or with 40 processing cost the agent could not achieve more than 89% for detection of coke using the biased saliency model. For analysis of the data difference in fitness functions and standard deviations should be noted.

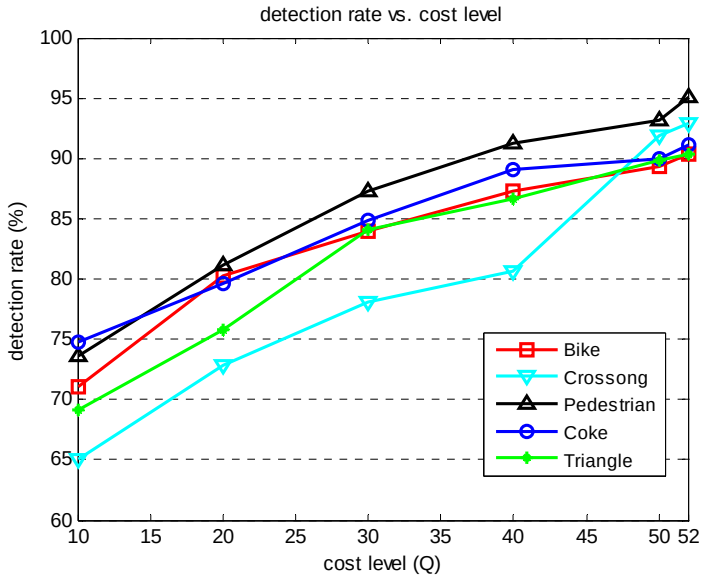


Figure 5.8 Mean detection of cost-limited case for traffic signs and objects over test sets.

Three first salient points generated by the biased saliency model (using the best individual in the final population in without-cost case) over 3 sample images of each object are shown in Fig. 5.9.

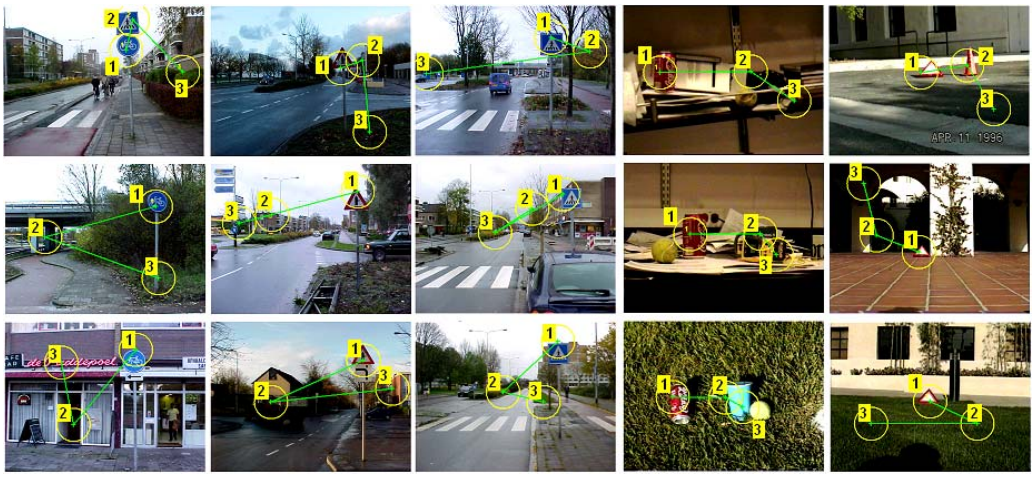


Figure 5.9 Three first attended locations of the biased saliency model over traffic signs and the objects. Numbers indicate the order of attended locations. Columns from left to right are for bike, crossing, pedestrian, coke and triangle objects.

Experiment III. Detection performance over disrupted images In this experiment, we compare the biased saliency model in without-cost case against basic saliency model and template matching which is a basic benchmarking method for object detection over disrupted images with several types of noises. In template matching, a window containing an instance of an object is slided over the image and correlation (or convolution) of each image region with the object is calculated. Template matching is very sensitive to image distortions like in-plane rotation and also to template selection. Gaussian, salt & pepper, speckle and motion blurred noises were selected because they simulate rainy and snowy weathers and movements in driving situations.

Achieved weights after CLPSO training in experiment II are used in this experiment over noisy test images. Three salient locations were proposed by each method as the most probable locations containing signs. Fig. 10 illustrates a sample image under typical noises and 3 locations proposed by the biased saliency model. Table 4 compares detection rates of the biased saliency model with template matching and basic saliency model.

Figure 5.10 Pedestrian sign detection over noisy images with biased saliency model. From left to right: Gaussian noise (μ = 0, σ = 0.5), salt & pepper noise (d=0.6), speckle noise with density d=0.5, motion blurred with a 20×30 window and correlation map over original noiseless image in template matching. White circle illustrates the most salient location and other 2 circles show less similar ones.

As results in Table 4 show biased saliency model performs better than the basic saliency model over all noises and have near the same computation time as it. While template matching has lower detection rates compared with biased saliency model, it is very sensitive to Gaussian noise and is more stable over other noises. However its computation time is much more than other two approaches. Template matching needs 3.5 times computation time more than the basic saliency model on average (3 times more than biased saliency model) on a computer with 1.4 GHZ Intel cpu and 512 MB ram memory.

Table 5.4 Mean detection rates and computation times of traffic signs over test images with 5 runs. Computation time is for detection of targets without noise. Numbers in parentheses are standard deviations.

	Without noise	Gaussian	Salt & pepper	Speckle	Motion blurring	Computation time
Method	-	(μ=0, σ = 0.5)	(d = 0.6)	(d = 0.5)	window size (20× 30)	(ms)
Bike						
TM	74.8(1.1)	31.5(2.1)	58.5(3.3)	79.5(2.3)	63.7(5)	292.1(20.4)
Basic Saliency	81.8(2.5)	63(4.3)	68.8(6.2)	83(5.5)	60.1(3.8)	86.23(12.2)
Biased Saliency	90.2(2.4)	78.1(3.1)	70.9(4.2)	86.1(5)	77.6(3)	98.4(10.5)
Crossing						
TM	75.5(1.6)	37.9(2.2)	64.2(2.7)	72.4(3)	75(1.9)	319.8(12.1)
Basic Saliency	78.2(1.8)	39.2(3.4)	59.2(4.9)	80.8(4.3)	86.4(3.3)	92.66(10.1)
Biased Saliency	95.8(2.7)	48.2(3.4)	62.2(4)	88.3(5.6)	93.1(3.2)	102.7(15.4)
Pedestrian						
TM	75.2(2.9)	25.5(2.7)	47.1(4.1)	75(3.9)	75.5(4.3)	315.5(14.3)
Basic Saliency	83.3(1.5)	51.7(4)	59.4(3.9)	71(6.1)	74.5(5.9)	90.27(12.5)
Biased Saliency	94.2(1.5)	77.1(2.5)	70.9(3.5)	85(3.5)	75.7(5.3)	106.9(14.2)
Average						
TM	75.1(0.3)	31.6(6.2)	56.6(8.7)	75.6(3.5)	71.4(6.6)	309.1(14.9)
Basic Saliency	81.1(2.6)	51.3(11.9)	62.4(5.4)	78.2(6.3)	73.6(13.1)	89.7(3.2)
Biased Saliency	93.4(2.8)	67.8(16.9)	68(5)	86.4(1.6)	82.1(9.5)	102(4.2)

5.4.2 Learning top-down spatial attention control

In many tasks, humans are biased (from their past experiences) toward specific spatial locations of their environment. For example when asking a person to look for a clock in a room, he will probably search on walls first instead of ceil. In this section, we propose a simple heuristic to reduce the saliency computation of the biased saliency model by using historical knowledge of the agent.

Experiment IV. Offline learning of task-relevance saliency map (TSM) Road traffic signs are typically placed either by the roadside or above roads. They provide important information for guiding, warning, or regulating the behaviors of drivers in order to make driving safer and easier. In this experiment, we used the priori assumptions on image formation like when signs are photographed from a driver's position or assuming that road is approximately straight. This leads to ignoring large portions of the image when looking for signs. Motivated by these restrictions,

we build a top-down task relevance map to consider such selectivity in space. Three salient locations were generated using biased saliency model for all training images for each traffic sign. A Gaussian mask was applied on each salient location to weight center locations more than surrounds. Then all maps were normalized and summed to form the final task-relevant spatial map shown in Fig. 11.

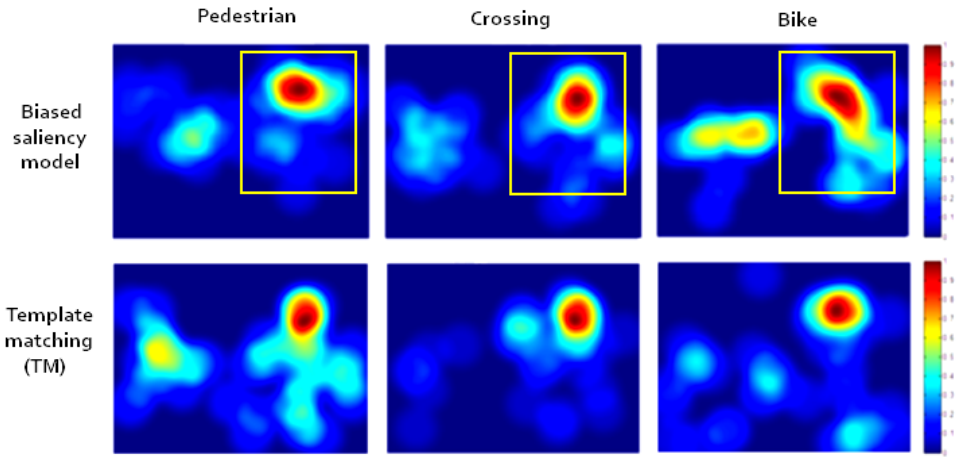


Figure 5.11 Task-relevant saliency map (TSM). Top row shows offline learned biased saliency map averaged for all images in the train set for pedestrian, crossing and bike signs from left to right. Three FOA's were generated for building this map. Bottom row shows the same but with salient locations computed with template matching method. As could be seen both maps seem to have high correlation but making such a map with template matching takes about 3 times more computation. Rectangles show the areas that saliencies were computed for detection of a sign.

An advantage of such offline top-down task-relevance saliency map is that saliency computation and object detection could be started first from spatial areas which have higher probability to contain an object and then to other areas. In order to use TSM for sign detection, saliency computation was limited in rectangular areas of Fig. 11. Over test sets of all three traffic signs, we were able to achieve the detection rates as before (Table 2), but about 3 times faster.

5.5 Discussions

The focus of this paper is on the learning phase of visual attention, where relevant feature values for a target are learned using several training images. Here the system automatically determines which features to attend in order to best separate a target from its surroundings.

An issue in revised saliency model is setting window size (n). In our experiments in this paper, we set values for this parameter experimentally based on extent of an object in scenes. We are looking for systematic determination of it. In general any fast surround inhibition operation which acts on a single scale can be used.

Our results show that color information has higher importance and information for discrimination of objects. Orientation features which have rough information about structure of objects are in the second rank. Intensity does not seem to have high information for object detection. Overall, it is hard to judge which scales are important in saliency detection from our

results. However, it seems that it depends on the size of the target object relative to the image size. With small number of training images, our biased system was able to detect objects in large number of unseen test images. As Table 2 shows our system has higher detection rates than the basic saliency model in without-cost case. In with-cost case while our system has higher detection rates compared to the basic saliency model it has lower cost. Over noisy images, our model performed better than template matching and basic saliency model. Its computation time is lower than that of template matching and slightly above the basic saliency model.

We also compared our attention system with two recent computational attention systems known as Neuromorphic Vision Toolkit (NVT) (Navalpakkam & Itti, 2006) and VOCUS (Frintrop, 2006). These systems were chosen to be compared with because they are the most similar models to ours and are both based on the basic saliency model. Average hit numbers of our method and these approaches are shown in Table 5 in terms of the parameters that has been reported in (Frintrop, 2006).

Table 5.5 Average hit numbers of our biasing approach versus two other biasing approaches

	target	# train images	# train images in VOCUS	# test images	avg. hit number		
					our method	NVT (Navalpakkam & Itti, 2006)	VOCUS (Frintrop, 2006)
campus map		9	2	7	1	1.2	1
fire hydrant		8	2	8	1	1	1
coke		45	5	59	1.6	3.8	1.3

As Table 5 shows our biasing method performed the same as VOCUS and NVT for fire hydrant detection. It was the same as VOCUS but better than NVT for campus map detection. It was better than NVT and slightly worse than VOCUS in detection of coke object. While sizes of train and test sets are small for the first 2 objects, it cannot be conclusive to judge which method is better than others from these 2 objects. However, since sizes of train and test sets are larger for the third object, it proves that our method outperforms NVT and competes with VOCUS. Performances reported in Table 5 are derived in without-cost of our algorithm. While performance of our approach is slightly lower than VOCUS, it proposes a framework to incorporate cost of features in object detection. For example for an agent, it might be acceptable to tolerate a lower detection rate but with smaller computation time while performing a real time task. In other words, our approach enables an agent with certain computational resources to gain maximum detection rate.

5.6 Conclusions and future works

In this chapter, we introduced a new method for learning top-down attentional control from offline training data. Our approach detects targets robustly and quickly in various scenes. It is

built over the basic saliency-based model of visual attention and biases it for synthetic saliency detection as well as natural object and traffic sign detection in cluttered scenes. It provides a method for quickly localizing object candidates to which computationally expensive recognition algorithms may be applied. Therefore, it provides a basis for robust and fast object recognition in computer vision and robotics. Since all feature channels and scales of the basic saliency model are not necessary for detection of an object, we put costs on computation of features of the model. The result was that, those channels which were not necessary and had not much effect on detection were not selected in the evolutionary process. Performance of our method was compared against basic saliency model as well as the template matching approach over noisy images.

A heuristic approach was also proposed for limiting the saliency computation over the most probable locations containing an object instead of the entire scene. For this, saliency maps from the previous experiences of the agent were added to form a top-down task relevance map. It was shown that this map reduces the computation time for traffic sign detection while having the same detection rate.

For our future research, we intend to learn top-down manipulations over the basic saliency model interactively and online using a reward and punishment mechanism. That way agent learns which objects to attend in each situation and then renders the image based on the top-down signals learned offline for that object. It would be also interesting to integrate the system into a robot control architecture enabling the detection of salient regions and goal-directed search in dynamic environments. In addition to the weights, there are also other parameters in the model which can be tuned, for example number of feature channels, number of scales in the image pyramid, window size of the surround inhibition operation and number of color and orientation channels. Making advanced use of object structure for biasing or modifying the basic saliency model can also be an interesting future work.

Chapter 6
Interactive Learning of Object-based Attention Control

Ali Borji, Majid N. Ahmadabadi, Babak N. Araabi

NIPS 2008 *Workshop* on Machine Learning Meets Human Learning
Under Review in:
Journal of Image and Vision Computing

Abstract. In this paper, we propose a biologically-motivated computational model for learning task-driven and object-based visual attention control in interactive environments. In order to do a task an artificial agent should be able to perform motor and perceptual actions at the same time. Our model consists of three layers. First, in the early visual processing layer, basic layout and gist of a scene are extracted. The most salient location of the scene is simultaneously derived using the biased saliency-based bottom-up model of visual attention. Then a cognitive component in the higher visual processing layer performs an application specific operation such as object recognition and scene understanding at the focus of attention. From this information, a state is derived in the decision making layer. Top-down attention in our model is learned by the U-TREE algorithm which successively grows a tree whenever perceptual aliasing occurs. Internal nodes in this tree check the existence of a specific object in the scene and its leaves point to states in the Q-table. Motor actions are associated with leaves. After performing a motor action, the agent receives a reinforcement signal from the critic. This signal is alternately used for modifying the tree or updating the action selection policy. A long-term memory component holds the bias signals of important task-relevant objects of the environment. Basic saliency-based model of visual attention is devised to consider processing costs of feature channels and image resolutions. To recognize objects, a recent and successful object recognition method, inspired by the hierarchical organization of the visual ventral stream, is used. The proposed model is evaluated on visual navigation tasks, where obtained results lend support to the applicability and usefulness of developed method for robotics.

Keywords: Top-down attention, Saliency-based model, U-TREE algorithm, Task-driven attention, Reinforcement learning, Q-learning, State space discretization

6.1 Introduction

Both biological and machine vision systems have to process enormous amount of visual information they receive at any given time. Attentional selection provides an efficient solution to this information overload problem by proposing a small set of scene regions to higher level and more computationally intensive processes like scene interpretation, object recognition, decision making, etc. In this regard visual attention acts as a front-end to a more complex vision system. Instead of processing all incoming visual information in parallel, the brain has evolved a serial strategy which explains its near real time performance in visual interactive environments.

Several physiological and psychological studies of attentional mechanisms in the brain have revealed that visual attention selects and gates visual information based on the saliency in the image itself (bottom-up) (Egeth & Yantis, 1997, Connor & Egeth, 2004) and on the prior knowledge about the scene (top-down) (Corbetta & Shulman, 2002, Desimone & Duncan, 1995, Chun & Wolfe, 2001). While bottom-up attention is solely determined by the image-based low-level cues, such as luminance and color contrasts, edge orientation and motion, top-down attention on the other hand is influenced by the task demands, prior knowledge of the target and the scene, emotions, expectations, etc. Bottom-up component of the visual attention is mainly examined by the early visual areas of the brain like LGN and V1 (Koch & Ullman, 1985, Li, 2001). Top-down attentional signals are largely derived from a network of areas in parietal and frontal cortex. Some of the involved areas include the superior parietal lobule (SPL), the frontal eye fields (FEF), the supplementary eye field (SEF) and the lateral prefrontal cortex in the region of the middle frontal gyrus (MFG). In daily life, these two mechanisms interact to control our attentional behaviors (Corbetta & Shulman, 2002, Kastner & Ungerleider, 2001). Besides acting in spatial domain by selecting spatial locations (Posner, 1980, Posner & Cohen, 1984), visual attention can also be directed to particular features such as color, orientation and direction of motion (Maunsell & Treue, 2006). It is also believed that attention selects objects rather than spatial locations (Kanwisher, J. Driver, 1992, Duncan, 1984, Kahneman & Henik, 1981).

Like humans and primates, artificial creatures like robots are limited in terms of allocation of their resources to huge sensory and perceptual information. That is mainly because of the serial processing mechanisms used in the design of such creatures which allows the processing of only a small amount of incoming sensory information. Since they are usually supposed to guarantee a short response time, attention is an efficient solution in robotics as in biological systems. In order to gain the maximum cumulative reward in the minimum time, agents should be able to perform perceptual and physical actions simultaneously. These perceptual actions are available in several forms like where and what to look in the visual modality. However, the main concern is how to select the relevant information, since relevancy depends on the tasks and the goals. In this study, we consider task relevancy of visual information and aim to extract spatial locations or objects which help the agent achieve its goals faster.

It is important that a solution for learning task-based visual attention control take into account other relevant and interleaved cognitive processes like learning, decision making, action selection, etc. Some evidences in this issue exist in both biology and engineering. It has been previously shown that attention and eye movements are context-based and task-dependent (Yarbus, 1967). Previous experiences also influence attention behaviors which indicate that attentional control mechanisms can be learned (Maljkovic & Nakayama, 1994). Some neuropsychological evidences suggest that human beings learn to extract useful information from

visual scenes in an interactive fashion without the aid of any external supervisor (Gibson & Spelke, 1983, Tarr & Cheng, 2003). In (Gray, 2003), it has been shown that attention is also affected by decision behaviors. By evaluating the consequences of our actions on the environment, we learn to pay attention to visual cues that are behaviorally important for performing a task. These findings are in accordance with a new and pragmatic view in Artificial Intelligence (AI) known as embodied and situated intelligence (Clark & Grush, 1999, Pfeifer & Bongard, 2006). It states that intellectual behaviors, representations, decisions, etc. are the product of interactions among mind, body, goals and the environment. There are also other supporting evidences in psychology claiming that human mind and intelligence have been formed interactively through an evolutionary process (Clark, 1999). Instead of attempting to segment, identify, represent and maintain detailed memory of all objects in a scene, there are evidences that claim our brain may adopt a need-based approach (Triesch et al., 2003), where only desired objects are quickly detected in the scene, identified and represented. Considering above evidences, in this work, we introduce a model to consider the influences of task, action, learning and decision making to control top-down visual attention of an agent.

In many real world situations, the environment is unfamiliar or not clearly defined. Moreover, required information and the optimal responses are not known at the design time. Therefore, fixed and predefined design of control strategies in such situations is less useful. Some complicated behaviors of humans like reading, writing, driving, etc. which need complex physical actions and attentions witness that such behaviors have been developed based on humans interaction with the surrounding world. Thus, interactive and semi-supervised approaches, e.g RL (Sutton & Barto, 1998), seem to be the most suitable techniques for learning top-down visual attention control and action selection strategies. Such learning mechanisms have the benefit of adapting the agent to dynamic, complicated and non-deterministic environments. In RL, agents learn action-values in each state by receiving a reinforcement signal from the critic. Another characteristic of RL methods is their ability of online learning which is required for interacting with stochastic and slowly changing environments. There are mathematical convergence proofs for these methods and they are biologically plausible (Seymour et al., 2004).

An agent living in an environment perceives information from its sensory inputs, takes actions and simultaneously receives reward signals from the environment. Because of the limited processing power, the agent must choose only a part of the information to attend. We intend to tackle this problem in this paper. Our proposed solution is built upon a sound and widely used bottom-up visual attention model proposed in (Itti et al., 1998). This model is based on the idea of saliency map, an explicit two-dimensional topographical map that encodes stimulus conspicuity or saliency at every location in the visual scene (Itti & Koch, 2001). Saliency in this model depends on the visual signal in a spatial location and its surrounding context. Bottom-up model is solely data-driven and simply selects some spatial locations without using any feedback mechanism or top-down gains. Some researchers have tried to add top-down capabilities to this basic model (Navalpakkam & Itti, 2005, Navalpakkam & Itti, 2006, Torralba, 2003), for instance by biasing it toward selecting specific objects. While such models are interesting, they have been partially successful to handle a limited category of tasks. Modeling top-down task-based influences on visual attention is difficult, since a general definition for a task does not yet exist. In this study, RL is used by the agent to learn how to select relevant objects and physical actions interactively in each state.

Particularly, in this work we introduce two contributions. First, we bias the basic saliency model of visual attention using a global optimization technique to find weight vectors with high discrimination power and low cost. This biased saliency model is used as a front-end to a higher vision unit in our model. Second, we use the U-TREE algorithm (McCallum, 1995) to dynamically discretize the visual state space when perceptual aliasing occurs. That way a binary tree is generated which is used for controlling top-down object-based visual attention control. Our model is inspired by the abstract findings from neuroscience and psychology.

In section 6.2, related researches are reviewed. Our proposed approach for learning task-driven visual attention control is explained in section 6.3. Experiments and results are shown in section 6.4. Section 6.5 brings the discussions and finally, section 6.6 summarizes and concludes the paper.

6.2 Related researches

Studies of visual attention fall into three categories. In the first category, researchers try to understand attentional mechanisms based on the psychophysical and behavioral experiments. Scan paths of eye movements, reaction times and percentage of correct responses are three example evaluation measures in this area. Second category involves neurophysiological studies. Single cell recording, fMRI and TMS are some techniques used to infer knowledge about the activities of neurons. Third category, involves mathematical modeling of visual attention mechanisms. Researchers in this category try to develop models for explaining behavioral and neuronal data. Models in this category are divided into two subcategories, known as connectionist and filter models. Connectionist models are based on single neuron models. By analyzing single neurons in networks, some aspects of complex behaviors could be explained. On the other hand, filter models, use higher level more abstract information gathered from experimental studies, such as Gabor filters as models of V1 neurons (Gabor, 1946, Hubel & Wiesel, 1965). They have been frequently used in machine vision community. The major goal of this area is to inspire from the main mechanisms and concepts of visual attention for developing artificial solutions, applications and frameworks for demanding problems in machine learning and artificial intelligence. Our work falls into this category and aims to propose solutions for real world applications.

In this section, we review visual attention studies which are directly related to ours, especially those which have considered learning aspects of visual attention in concert with decision making. First we review some existing hypotheses and viewpoints on visual attention mainly derived from behavioral studies and then bring some successful approaches from AI for learning attention control.

An important evidence from biology reported in (Maljkovic & Nakayama, 1994), states that attention could be learned by past experience. In a behavioral task, human subjects were supposed to answer a question about a quality of a specific visual item in a synthetic visual search scene. Subjects had lower reaction times when quality of the object stayed the same during successive trials. This study shows that subjects developed a memory during the task. A modeling work trying to explain such behavioral data is done in (Mozer et al., 2006). They have proposed an optimization framework to minimize an objective function which is a sum over the reaction time in each state weighted by the probability of that state to occur. Then using a Bayesian Belief Network (BBN), they solved that minimization problem. These results encourage using a learning approach for attention control in AI.

Some RL studies have previously been proposed for modeling top-down visual attention control in humans. Since eye movements have high correlation with overt visual attention, these studies have tried to explain eye movement data. Bottom-up models could not explain much of that data because they do not consider top-down effects. In (Reichle & Patryk, 2006), RL is used for modeling the behavior of an expert reader by predicting where eyes should look and how long they should stay there for achieving best comprehension from the text. This work is interesting as it models attention as an optimization problem and then uses RL to solve it. Another model of human eye movements is proposed in (Sprague & Ballard, 2005) that directly ties eye movements to the ongoing demands of behavior. The basic idea is that eye movements serve to reduce uncertainty about environmental variables that are task relevant. A value is assigned to an eye movement by estimating the expected cost of the uncertainty that will result if the movement is not made. If there are several candidate eye movements, the one with the highest expected value is chosen. The model is illustrated using a humanoid graphic figure that navigates on a sidewalk in a virtual urban environment.

RL has also been used for deriving visual attention policies for mobile robots. In (Fritz et al., 2004, Paletta et al., 2005), a 3 step architecture is proposed for an object recognition task. First, it extracts potential focuses of interest (FOI) according to an information theoretic saliency measure. Then it generates some weak object hypotheses by matching the information at the FOI's with codebooks. The final step is done using Q-learning with the goal of finding the best perceptual action according to the search task. In (Gonic et al., 1999.a), two approaches are proposed in a robotic platform with neck, eyes and arms for attention control. The first approach is a simple feedforward method which uses back-propagation learning algorithm while the second one uses reinforcement learning and a finite state machine for state space representation. The robot has 3 types of actions: attention shift, visual improvement and haptic improvement. Their results confirm that the second approach generates a better performance in terms of finding previously observed objects even with fewer movements in head and neck and also in attention center shift. In (Gonic et al., 1999.b), another robotic platform containing articulated stereo-head with 4 degrees of freedom is presented which can select the region of interest, perform attention shift with saccadic movements, build a map out of the environment and update it according to current observation. The main characteristic of this approach is that it observes the whole environment and selects one region to attend according to its saliency map. In fact, here, attention control has two steps: first, coarse eye movements and then more precise iterative adjustments around the first points. The termination condition of this process is reaching to a maximum correlation among what it finds and what it expects. Due to its simple search application, this work has not confronted to many real world challenges.

An approach for learning gaze control for a mobile robot is proposed in (Minut & Mahadevan, 2001), which proposes a model of selective attention for visual search tasks. Their model is implemented using a fixed pan-tilt-zoom camera in a visually cluttered lab environment, which samples the environment at discrete time steps. The agent has to decide where to fixate next merely based on visual information, in order to reach the region where a target object is most likely to be found. The model consists of two interacting modules. In the first module, RL learns a policy on a set of regions in the room for reaching the target object, using an objective function which is the expected value of the sum of discounted rewards. By selecting an appropriate gaze direction at each step, this module provides top-down control in the selection of the next fixation

point. The second module performs "within fixation" processing, based exclusively on visual information. An interesting point with this work is that it has incorporated learning where to look in a visual search task. Unfortunately, no suggestions for generalizing this method have been made. Another advantage of this work is its implementation on a working robotic agent.

A bottom-up visual attention model known as saliency-based model which is an extension and implementation of an earlier model of visual attention introduced by Koch and Ullman (Koch & ullman, 1985) is proposed in (Itti et al., 1998). This model is based on the saliency concept and mimics the overall structure of the early visual system for detecting the locations which convey more visual signals and are different from their surroundings. For example a red dot is more discriminant than a blue dot in a white background containing blue dots. Simplicity and little computation are the two main advantages of this model. It has been continuously updated and is the basis of the newer models. It has also been used to explain behavioral data on simple synthetic and static search arrays to dynamic natural stimuli like movies and games (Peters & Itti, 2008). In this study we use this model to build our top-down attention control system upon it. A detailed description of this model is shown in section 3.1.1. In (Navalpakkam & Itti, 2005), a task-based model of visual attention control is proposed based on the saliency model to add top-down capabilities to it. Given a task definition in the form of keywords, this model first determines and stores the task-relevant entities in working memory, using prior knowledge stored in a long-term memory. It then attempts to detect the most relevant entity by biasing its visual attention system with the entity's learned low-level features. It attends to the most salient location in the scene, and attempts to recognize the attended object through hierarchical matching against object representations stored in the long-term memory. It updates its working memory with the task-relevance of the recognized entity and updates a topographic task relevance map with the location and relevance of the recognized entity. Instead of a predefined definition in the form of keywords or a sentence for a task we would like to learn manipulations over the basic saliency map by rewards and punishments that the agent receives. This approach is more general because it allows the agent to interact with the environment and to find its own way of achieving a goal.

In (Jodogne & Piater, 2007), Jodogne et al. have presented a framework known as RLVC (reinforcement learning of visual classes) for learning mappings from images to actions by interacting with the environment. RLVC consists of two interleaved learning processes: 1) an RL unit which learns image to action mappings and 2) a binary image classifier which incrementally learns to distinguish visual classes when perceptual aliasing occurs. The classifier acts like an attention tree by checking whether a specific SIFT feature is present in the image or not. RLVC is the extension of a previous seminal work known as U-TREE algorithm (McCallum, 1995) to visual domain. The main idea behind both approaches is that state-space is incrementally discretized whenever an aliasing occurs. The main drawback with the RLVC is its exhaustive search overall image for computing the SIFT features which contradicts the philosophy of existence of visual attention. In our method, like RLVC, an action-based decision tree is constructed. We use the basic idea of U-TREE algorithm for refining the representations of the agent. Learning representations result in an optimal decision tree, in sense of maximizing the reward of the agent, for attention control.

There are also other computational models of visual attention. Deco and Schurmann (Deco & Schürmann, 2000) modulated the spatial resolution of the image based on a top-down attentional control signal. Tsotsos et al. (Tsotsos et al., 1995) used local winner-take-all networks (WTA)

and top-down mechanisms to selectively tune model neurons at the attended location. Making extensive use of feedback and long-range cortical connections, Hamker (Hamker, 2005.a, Hamker, 2005.b) modeled the interactions of several brain areas involved in processing visual attention, which enables them to explain some physiological and behavioral data. Sun and Fisher (Sun & Fisher, 2003) have developed a framework for object-based attention using "groupings". Presented with a manually preprocessed input image, their model replicates human viewing behavior for artificial and natural scenes. Like this work, attention in our model selects objects instead of spatial locations or features.

Our work extends the previous works in this area by proposing a top-down object model for attention control which attends and recognizes to those objects which are important in decision making. The running power of our method is the reinforcement signals the agent receives from the environment.

6.3 Proposed model

An agent working in an environment receives information momentarily through its visual sensor. It should determine what to look for. For this we use RL to teach the agent simply look for the most task-relevant and rewarding entity in the visual scene. Our model consists of three layers: early visual processing, higher visual processing and decision making layers as shown in Figure 6.1. In order to keep the model simple but functional, only important interconnections among components are considered. This model is a extension of our earlier model to visual interactive environments (Borji et al., 2008.a).

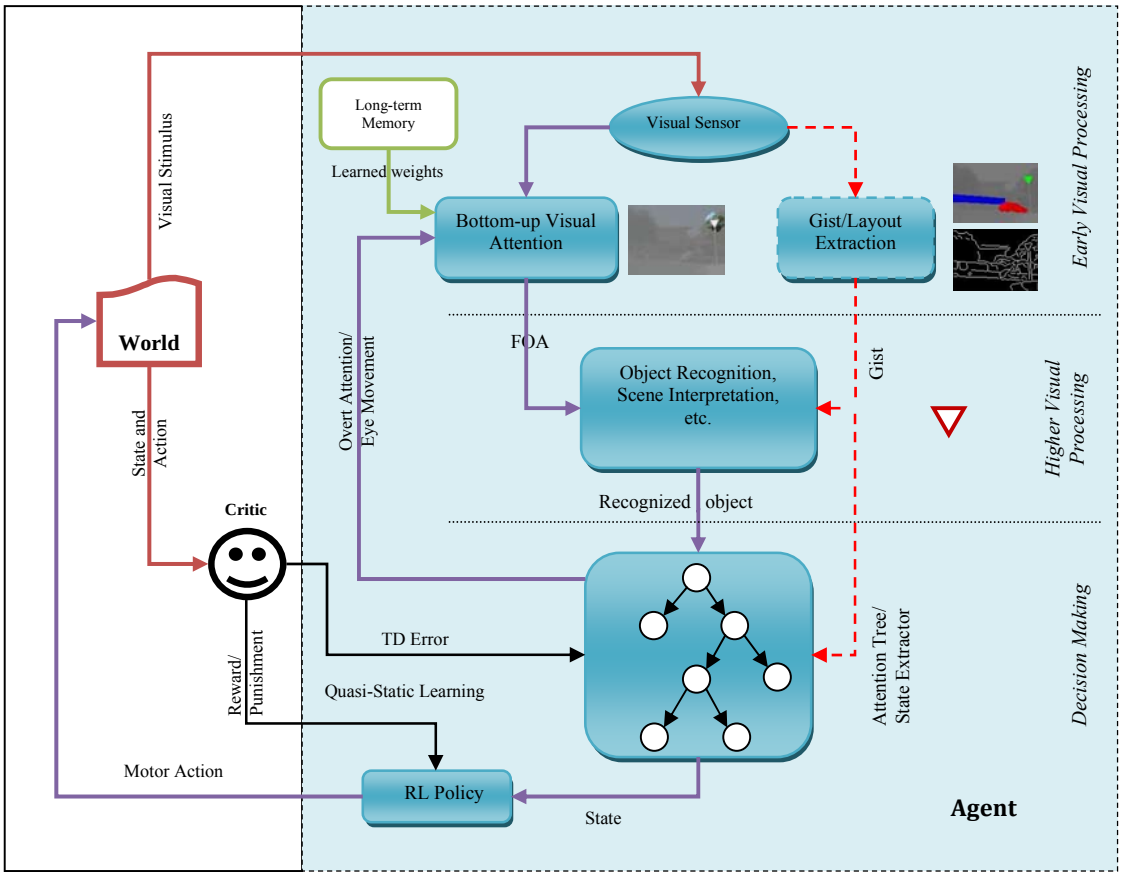


Figure 6.1 Proposed model for learning task-driven object-based visual attention control. The model consists of three layers, early and higher level visual processing layers and decision making layer. Biased saliency model selects a spatial area, higher vision is focused on this area to recognize the object at that location and then a state is extracted by traversing the attention tree. A reinforcement signal in response to the action elicited at that state is used alternatively for refining the tree or updating the action selection policy. Long-term memory component is used to save the offline learned biasing weights of the saliency model. Gist component (shown in dashed lines) extracts rough information and is subject to our future works. The generated tree has nodes with out-degree either 0 or 2.

An example scenario of the model is as follows; see Figure 6.1. Captured scene of the environment through agents' visual sensor undergoes a biased bottom-up saliency detection operation and a focus of attention area is selected. Gist and layout of the image are also extracted by the gist/Layout extractor unit at the same time. Both the most salient location and the overall sketch of the image are then transferred to the higher vision unit. For reducing the computational complexity, higher visual processes (like object recognition) are only targeted at the focus of attention (FOA). Next state of the agent is determined by the attention tree. Motor actions are associated with the leaves. Outcome of this action over the world is evaluated by a critic who is aware of the model of the environment and a reinforcement signal is fed back to the agent to update its internal representations (attention tree) and action selection strategy in a quasi-static manner. Following subsections discuss each layer of the model in detail.

6.3.1 Early visual processing layer

Two components in this layer process the captured image in parallel: bottom-up visual attention and gist/Layout extractor components. The bottom-up attention component first determines the saliency of all spatial locations and then selects the most salient location at the FOA. Instead of biasing the saliency model online at the time of interaction with the world, here an offline approach is adopted. This is mainly because parameters of the model are continuous valued, therefore an online mechanism for learning top-down gains greatly degrades the RL convergence speed. The image information at the FOA is then transmitted to the higher vision unit for further processing. Gist extractor unit in this layer performs a rough categorization of the image (Oliva, A. Torralba, 2006). A long-term memory component is used by the bottom-up attention component and holds the information of important objects like biasing weights of the saliency model. The early vision unit is the artificial counterpart of the V1 and V2 cortical areas in the visual cortex. (what / where pathways)

Bottom-up attention component Saliency model in presence of no bias selects a spatial location in a feedforward manner. In (Navalpakkam & Itti, 2006), the basic model is biased toward a specific object by maximizing the SNR ratio of the object to the whole image. In this section we modify the basic model and then use a global optimization technique to learn biasing weights of the saliency model offline. Instead of only finding the appropriate weights, we also incorporate processing costs of the feature channels to force the optimization process to choose the feature vectors with high detection rate and low cost (Borji et al., 2008.a). Revised saliency-based model of visual attention was explained in chapter 5. We use the results of biasing for object detection to build our top-down attention paper in this chapter.

Gist/Layout extractor component Gist in psychophysical terms is the ability of people to roughly describe the type and overall category of an image after only a very brief presentation (about 100ms) and to use this information to guide subsequent target searches. It indicates that we are not blind outside our focus of attention and information outside the FOA is also processed to the extent less than the attended area. Layout extraction is done using basic image processing operations like edge and contour detection. Gist in our model could be used by for example searching the attention tree more efficiently by bypassing some sub-trees. So, it helps the agent to determine its state faster. In other words, gist in our model like as a heuristic which directs attention. It is calculated very fast in a bottom-up manner and helps top-down control of attention. This information is necessary for complex cognitive operations in the upper layers. The low computational cost of extracting gist features facilitates effective usage in applications such as content-based image retrieval and robot vision localization. Some models have been previously proposed in the literature for calculating the gist of a scene by Oliva and Torralba (2006), Reninger and Malik (2004), Siagian and Itti (2007) and Torralba et al. (2003). A comparison of these methods is provided in (Siagian and Itti, 2008).

6.3.2 Higher visual processing layer

This layer acts upon the information extracted by the early visual processing unit. Operations in this unit are more sophisticated and give the agent a more cognitive understanding of the

image/scene, useful for decision making in the decision making layer. Example operations in this unit are object recognition, scene classification, background subtraction, etc. Processes in this unit are dependent on the environment and the tasks of the agent. Other processes like emotions, expectations, internal states, etc. which have already been proved to be important in deriving attention, are also expected. This layer corresponds to cortical areas at the end of visual ventral stream like V4, IT and PFC. Theoretically any object recognition method can be used in this layer. However to make the model to sound more biologically inspired, we use the standard model of object recognition for recognizing objects at the focus of attention.

Standard model of object recognition (HMAX) It is claimed that visual attention and object recognition are tightly linked processes in human perception. Our model proposes how object recognition mechanism could be implemented over the bottom-up component and is a new way of explaining the interaction of these two components. This interaction happens under the influence of top-down and task-based control signals. Here, we briefly explain the standard model of object recognition, a hierarchical model which closely follows the operations and findings from the early and late visual system. It will be used for object recognition in visual navigation tasks in the section 4. Note that as in the saliency model HMAX is also trained offline and is used online when performing a task.

The hierarchical model of object recognition in cortex by Riesenhuber, T. Poggio (1999, 2003) mimics the hierarchical organization of the primate visual ventral stream (the feedforward pathway). The model consists of four alternative layers of Simple (S) and Complex (C) units. S1 units extract local orientation information from the input image by convolution with Gabor filters, at 4 orientations at different scales. S1 activity is pooled over local spatial patches and 4 scale bands using a maximum operation to arrive at C1 complex cells. In the next stage, activities from C1 cells with similar positions but different orientation selectivities are combined in a weighted sum to arrive at S2 composite feature cells that are tuned to a dictionary of more complex features. The dictionary we use in this section consists of all possible combinations of the four cardinal orientations in a 2×2 grid of neurons, i.e., $(2 \times 2)^4 = 256$ different S2 features. In a final non-linear pooling step over all positions and scale bands, activities of S2 cells are combined into C2 units using the same maximum operation as used from the S1 to the C1 layer. The activity patterns of the 256 C2 cells feed into view-tuned units (VTUs) with connection weights learned from exposure to training examples. VTUs are tightly tuned to object identity, rotation in depth, illumination, and other object-dependent transformations, but show invariance to translation and scaling of their preferred object view. In their selectivity to shape, S1 and C1 layers are approximately equivalent to simple and complex cells in areas V1 and V2, S2 to area V4, and C2 and the VTUs to areas in posterior infero-temporal cortex (PIT) with a spectrum of tuning properties ranging from complex features to full object views. C2 features of the model are used here for classification. In our experiments we used a MATLAB® implementation of the model[3]. Model parameters used in our experiments are the same as in (Serre et al., 2007). For more detailed information on the HMAX model, the interested reader should refer to Serre et al. (2007, 2004, 2005). HMAX has been successfully applied to some computer vision and pattern recognition problems (Serre et al., 2007, Borji et al., 2008).

[3] Implementation of the HMAX model could be downloaded from: http://riesenhuberlab.neuro.georgetown.edu/hmax.html.

In order to train the standard model, first a binary mask (50×50) is placed at the location of the object in the image and then object areas are extracted. For each class a binary SVM classifier (Vapnik, 1995) was trained using 20 samples of the target class (positive patterns) and 20 samples from each other class (negative patterns). The trained classifier was later tested over the 10 remaining positive and negative patterns of all classes. Classification using the C2 features resulted in 87% (±3.2%), 93% (±2.75%), 91.28% (±2.8%), 94.6% (±1.4%) and 83% (±4.2%) recognition rates for pedestrian, crossing, bike and triangle signs and the coke object respectively. Reported results are averaged over 5 runs with random training patterns.
This learned offline knowledge will later be used by the agent in order to recognize an object at the focus of attention.

6.3.3 Decision making layer

The core of our model is the decision making layer where visual attentions and representations are learned. This layer controls both top-down visual attention and motor actions. Top-down attention is equivalent to overt attention which in humans means that attention is relocated with eye movements. The motor actions are executed by the body of the agent and affect the world. State extractor unit (attention tree) in this layer, derives the state of the agent based on the cognitive information it receives from the higher vision unit. The learning approach is an extension of the U-TREE algorithm to visual domain (McCallum, 1995).

Based on the derived state, the agent must choose to either do a motor action or perform another perceptual action which here means looking for another object to clarify its state. The agent traverses the attention tree until it reaches a leaf node which determines its state in Q-table. Motor and perceptual actions affect the world and the bottom-up visual attention component, respectively. An evaluation of the motor action over the world is fed back to the agent by the critic in form of a reinforcement signal. This signal is used by the agent for updating its policy. Since there are two interleaved parameters in the model, attention tree and action selection policy, learning both of them if possible may take a long time. In this study, we adopt a quasi-static approach for learning the attention tree and image to action mappings. Each time reinforcement signal from the critic is used for either modifying the tree or updating the Q-table. This unit corresponds to decision making areas in the brain like LIP and PFC cortices.
Details of the algorithm for learning top-down attention tree are explained in the next section.

Learning attention tree An efficient method to implement attention and state space construction is by means of *tree* data structures. Such structures are interesting because they allow learning representations and attention control at the same time. Visual discretization is achieved via expanding the attention tree whenever perceptual aliasing occurs. Such refinement is performed to increase the cumulative reward of the agent in each time step t. (equation 6.1).

$$R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1}, \; where \gamma \in [0,1] \tag{6.1}$$

Each internal node of the tree checks whether a specific object exists in the scene or not. The generated tree for attention control is a binary tree, since an object checked in an internal node of the tree either exists or not in the scene (Figure 6.2). When an image is presented to the agent,

attention tree is sequentially traversed from the root until a leaf node is reached (*traverseTree()* function in table 3).

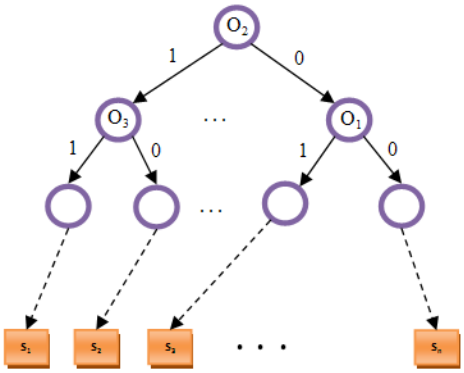


Figure 6.2 Binary attention tree. Internal nodes of the tree check the existence of a specific object in the image. Leaves of the tree represent the world and correspond to states in the Q-table.

Attention tree is incrementally built in a quasi-static manner in two phases (iterations). 1) RL-fixed phase and 2) Tree-fixed phase. In each phase of the algorithm external feedback of the critic (a scalar reward) is used to alternatively update the policy or refine the leaves with aliasing.

Initially a tree with a single node is created and all the images are mapped to that node. It is obvious that such a single state is not enough in many cases and therefore aliasing occurs. Then, algorithm breaks that node into two leaves based on some gathered experiences beneath it. Algorithm starts with a single node in the RL-fixed phase and then moves to the Tree-Fix phase and so on. In each Tree-fixed phase, RL algorithm is executed for a number of episodes according to the learned policy from the previous phase by following an ε-greedy action selection strategy. In this phase, tree is hold fixed and the derived quadruples (s_t, a_t, r_{t+1}, s_{t+1}) are only used for updating the Q-table according to Q-learning formula (equation 6.2) (Watkins & Dayan, 1995).

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a)\right) \tag{6.2}$$

where, s_t is the state of the agent at time t. The agent performs action a_t, receives reward r_{t+1} and enters state s_{t+1}.

State discretization occurs in the RL-fixed (Tree-update) phase. In this phase, gathered experiences by the agent are used to refine leaves of the attention tree with aliasing. An important point to be considered here is that the agent only accesses the environment through its visual sensor (e.g. its CCD camera). Therefore, in order to determine its state in any position, the agent should capture a scene in that position and then traverse its attention tree from the root node down to a leaf.

Measuring aliasing After each Tree-fixed (RL-update) phase, the learned tree is refined by expanding the states (leaves) with perceptual aliasing. In order to estimate aliasing, a number of patterns should be accumulated under a leaf node (*gatherMem()* function in table 6.1). To have a better estimation for aliasing, we only refined those leaves with the memory size greater than a

threshold value (*memThreshold*). Memory is gathered by the agent performing some episodes with the learned action selection policy from the previous Tree-fixed phase. An experience under a node with state s_t is $([o_1, o_2, \dots, o_n], a_t, \Delta t)$, $o_i \in \{0, 1\}$, where o_i is 1 if object o_i exists in the scene. As in the Tree-fixed phase, an image is captured, attention tree is traversed in order to find the perpetual state, appropriate action is performed and a reward is received. A prominent measure of perceptual aliasing in a state (leaf node) is the TD error (also known as Bellman residual) and is derived from the Q-learning formula as shown in equation 6.3.

$$\begin{aligned} Q(s_t, a_t) &= (1-\alpha)Q(s_t, a_t) + \alpha(r_{t+1} + \gamma\, max_a\, Q(s_{t+1}, a)) \\ &= \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right) + Q(s_t, a_t) \\ &= \alpha\Delta t + Q(s_t, a_t) \end{aligned} \tag{6.3}$$

Δt, is the TD error of state s_t with respect to action a_t. In order to detect aliasing, all patterns under a node are clustered according to their physical actions and then if any of these clusters has a variance in Δt's greater than a threshold (*aliasingThreshold*), then that node has aliasing at least with respect to one action. Therefore, equation 6.31 reduces to equation 6.4, because $Q(s_t, a_t)$, is the same for all clustered patterns under a node.

$$\Delta t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \tag{6.4}$$

When the RL algorithm converges, then the Q-values do not change any more and Δt's vanish and there is no further aliasings.

Tree refinement In order to refine the attention tree, aliased states are expanded into two leaves. Tree modification (refinement) is done by selecting the object which mostly reduces the variance in Δt of patterns in the memory of a leaf according to equation (6.5). In (Asadpour et al., 2006), splitting measures for U-TREE are compared and is shown that variance is a more effective measure in terms of generating tress with smaller number of states and less computation time than other measures.

$$\begin{aligned} o^* &= armin_o \left(var\{y\} - \left(\frac{|L_+|}{|L|} var\{y_a| +\} + \frac{|L_-|}{|L|} var\{y_a| -\}\right)\right) \\ &= armax_o \left(\frac{|L_+|}{|L|} var\{y_a| +\} + \frac{|L_-|}{|L|} var\{y_a| -\}\right) \end{aligned} \tag{6.5}$$

where y_a is the set of all memory items with action a, $y_{a|+}$ ($y_{a|-}$) is the set of memory items with (without) object o and action a. Sizes of these two sets are $|L_+|$ and $|L_-|$, respectively.

Maximization is done over all pairs of objects and actions. An object which minimizes the variance also has to cluster the images under the node into two populations with significantly different distributions. Here we used t-test for comparing these two distributions. If the power of the t-test is below 0.05, then two distributions are considered to be significantly different. When expanding a node, an object is selected which has not been already used in the path from this

node to the root. When a leaf is created, a corresponding state is added to the Q-table (resp. for deletion) and the Q-values are uniformly initialized.

Tree pruning Proposed algorithm constructs an attention tree in a greedy manner which usually leads to overfitting. Therefore solutions should be designed to overcome overfitting by either periodic tree restructuring or pruning. Two heuristics are introduced in the following.

Consider two types of nodes: 1) nodes having leaves with the same best actions learned for them 2) nodes with at least one child with no memory. Leaves of the nodes in the first category are removed and their action is assigned to their parent node. A node in the second category is removed and its child with memory is substituted with it. These two heuristics are recursively done from bottom to top of the tree until no node satisfies one of these conditions. The whole algorithm for learning image-to-action mapping is summarized in the pseudocode of table 6.1.

6.4 Experimental results

Performance of the proposed model is analyzed in this section over a visual navigation task which closely parallels real-world scenarios. First, it is assumed that motor actions are already known and only states of the environment should be discovered. This assumption is later relaxed and both attentions and motor actions are learned for safe navigation. The agent has no access to its (x, y) position and only receives images in each situation (taken by its CCD camera). At each state, the agent has three possible motor actions: Forward (F), Turn Left (L) and Turn Right (R). It can attend to one of several objects each time. For checking the existence of an object, when moving in its attention tree, the agent applies the learned biasing signals for that object to the observed scene (Long-term memory component in the model). Then the most salient region of this biased saliency map is transferred to the corresponding trained binary SVM classifier (associated with the attended object), and based on a positive or negative answer from this classifier moves in the attention tree until it reaches a leaf node determining its state.

6.4.1 Visual navigation task

In this experiment, the agent is supposed to learn how to navigate safely in a simulated driving environment. The agent uses its offline learned knowledge interactively. Map of the route, consisting of 4 positions, is shown in Figure 6.3. The agent captures 360 × 270 RGB color images. It is assumed that the agent knows which objects are in the scene when it observes it. This information is only used for refining the aliased nodes and not for state determinations.

Optimal policy is not unique in this map. They are shown with red characters {F, R, L} besides each state in Figure 6.3. Note that, two actions might be the best for a state. Since there are 3 positions (minus Goal) and 4 sides per each one, there are 3×4 =12 states in the environment. The agent in each of the 12 states captures a scene containing either one or more objects. In each episode of RL, the agent is placed randomly in a position of the map. An example assignment of objects (Pedestrian (A), Crossing (B), Bike(C) and triangle (D)) to states is shown in figure 6.3. For each combination, we created 5 possible images with positions of objects randomly selected. In this experiment, we assumed that the agent is always capable of detecting and recognizing the objects in the scenes. Actually, scenes were selected in such a way to give the agent a deterministic behavior. It means that when the agent applies the leaned biases for a specific object to the bottom-up attention model, it can correctly detect and recognize that object

in the scene using only the first saccade. Examples of the natural scenes with embedded traffic signs are shown in figure 6.4. Figure 6.5 illustrates a sample image when the agent selectively attended to one of two different signs.

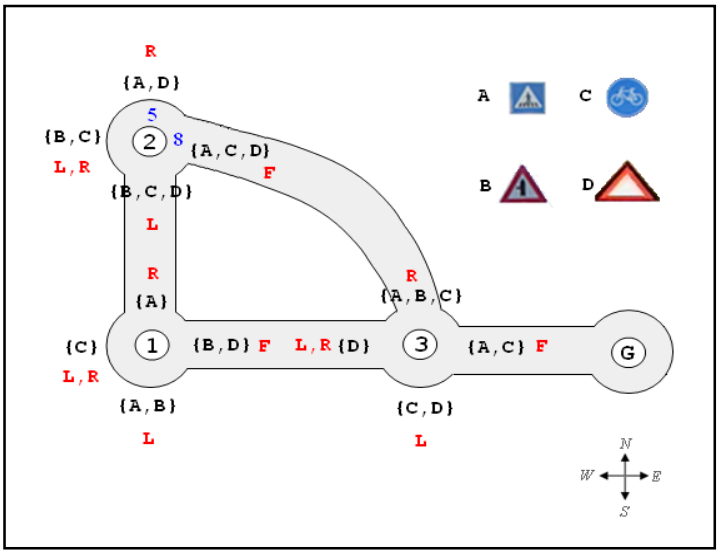


Figure 6.3 Simple navigation map. Goal state is shown with capital letter G. The agent can go forward, turn left or right and can only attend to one object in the scene each time. It should reach the goal state as fast as possible.

States of the environment are numbered in this way: $(x\text{-}1)\times 4 + y$, where x is the position number (in the central circle) and y is the head direction of the robot (in the coordinate system at the bottom-right of figure 6.3). $y = 1, 2, 3$ *and* 4 for North, West, South and East directions, respectively. For instance, state number of north side of position 2 is 5 and for its east, it is 8. Note that the best actions in the image are with respect to the head direction of the agent. For example, when head of the agent is toward east in the map, south lies in its right side.

Motor actions known and fixed Since the navigation task, is a coupled task, it demands learning both state representations and motor actions. In this section, we aim to solve the case in which motor actions for states are already known. Therefore it only remains to learn the states and their associations with the motor actions. Then we relax this assumption and use our model for learning both representations and their associated motor actions in next section.

Figure 6.4 Sample scenes with embedded objects. Five images for each state were randomly selected. Positions of signs were also determined randomly. Images were selected in such a way that both saliency model and HMAX could detect and recognize them correctly at the first saccade.

Figure 6.5 Selective attention to two different objects in the same scene. Only first attended location is shown.

Table 6.1. Algorithm for learning top-down object-based attention control

```
main ()
 tree = Create a tree with a single node                  //all input images are mapped to this single state at the start
 Repeat

     RL- Fixed phase                                      // in this phase only tree is modified
      for  i = 1 to maxEpizodes
           I_t  = take an image
           s_t = traverseTree(tree, I_t)                  // s_t is a leaf node
           [s_t+1 r_t+1 a_t] = performAction(s_t)         // based on an action selection policy choose action a_t,
                                                          //   go to state s_t+1 and get reward r_t+1
           Δt  = calcDelta (s_t, s_t+1, r_t+1, a_t)       // Δt is the TD error according to equation 11.
           mem = gatherMem(a_t, I_t, Δt)                  // this item is saved for state (node) s_t
      end                                                 // end for i

      for j=1 to size(tree.leaves)
       if  size(mem) >  memTreshold                       // expand the nodes with aliasing
           if  checkAliasing(s_t)
                tree = modifyTree(tree, s_t)
      end
     pruneTree(tree)                                      // end for j
                                                          // tree is pruned in some intervals (iterations)
     Tree- Fixed phase
      for i = 1 to maxEpizodes                            // in this phase only Q-table is updated
           I_t = take an image
           s_t = traverseTree (tree, I_t)
           [s_t+1  r_t+1  a_t] = performAction(s_t)
           Q-table = updatePolicy(Q-table, s_t)
      end
                                                          // end for i
  Until (no more aliasing) or (maximum iterations is
reached)                                                  // repeat
traverseTree (tree, I_t)
  node = tree.root;
  while node.childSize > 0
     flag =  attend and recognize the object at node
     if  flag
          node = node.leftChild
     else
          node = node.rightChild
     end
   end                                                    // while
  return node
checkAliasing(s_t)
  for action a ∈ A                                        // A  is the set of all motor actions
     mem(a) = all mem with action  a  under s_t
     var(a) = calcVariance(mem(a))                        // of mem(a)
     if var (a) > aliasingThreshold
              return true;
  end                                                     // end for action a
  return false;
modifyTree(tree, s_t)
   for action a ∈ A
     mem(a) = all mem with action  a  under s_t
     for  object o ∈ O                                    // O is the set of all possible objects the agent might observe in the scene
        choose the object which reduces variance the most
        according the equation 13 and also partitions the
        memory into two populations with non-equal
        distributions  (using t-test )
     end                                                  // for o
   end                                                    // for a
```

The agent receives both immediate and delayed rewards. It starts with a single *Null* state and all captured images are mapped to that state. For a randomly captured image, the agent must choose between doing a motor action and attending to another object to clarify its state (to increase the chance of correct classification of this image). In the *Null* state, the agent is only allowed to do attentions (and not motor actions), since it is clear that more than one state is needed for representing the environment. This causes the algorithm to generate other states. In each state, the agent performs either an attention or a motor action using ε-greedy action selection policy. If predicted motor action by the agent for this state is optimal (best), it receives a big reward of +100 and otherwise it gets -100 punishments. If it attends to an object, it receives a punishment which is the cost associated with that object. Then, the agent updates the Q-table entry for this state and moves to the next state. If the next state does not exist in the Q-table, then it is created and initialized uniformly. This process is repeated for some episodes (maxEpisodes). Flowchart of figure 6.6 illustrates this process.

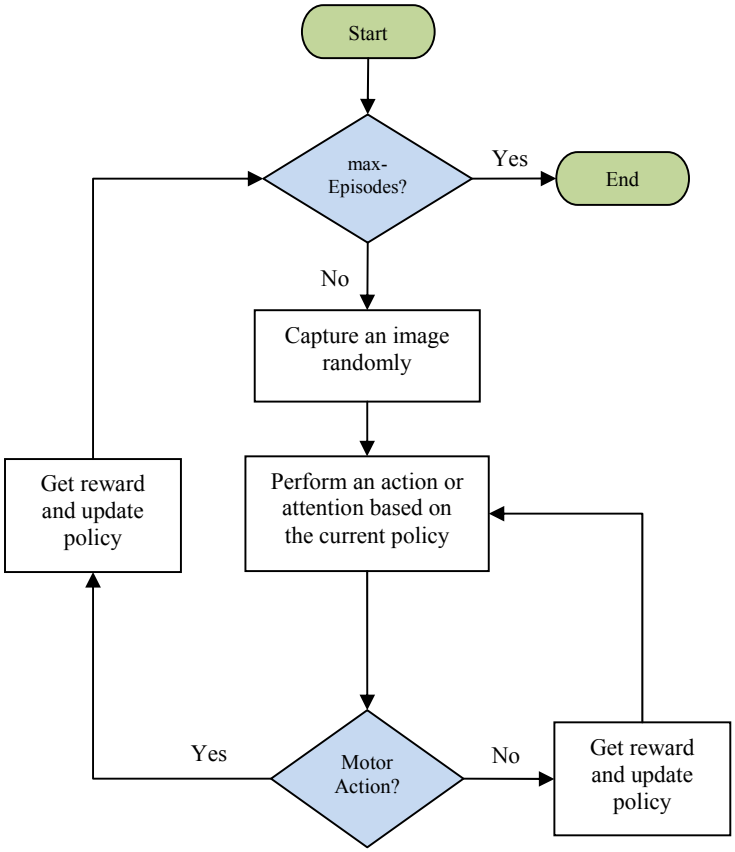


Figure 6.6 Flowchart of the algorithm for learning attention control when motor actions are known in advance. State of the agent is determined by the objects it observes. The agent finally learns to do a motor action or to attend to another object in each state when it is presented with an image.

Cumulative average rewards and average costs for this experiment are shown in figure 6.7. Results are compared with the situation when the agent has the full observation (full attention). In this case, the agent checks the existence of (attends at the same time to) all objects in the scene in

order to find its state. A '1' means that the attended object exists in the scene and '0' means it does not. Since the response for attending to an object is either 1 or 0, state space (with 4 objects) in full attention case has 2^4 states.

As shown in figure 6.7, both attended and full-observation cases, converged to the same average reward in iteration 2000 (bottom panel). The reason why both curves are below +100 is because of the punishments the agent receives when performing the task. Top panel of figure 6.7 shows the average cost of both approaches. When the agent attends to all objects, it always has the cost of (-10 = - 4 - 3 - 2 - 1). So, its average cost is always -10. When it attends to one object at a time, it has a higher average cost at first, but as long as it learns to do the task (and to attend), its average cost reduces and finally reaches a value above -10. To find whether a specific object is available in the scene or not, the agent first applies learned top-down gains for that object to the image, then a square area (50 × 50 pixels) is sent to the SVM classifier corresponding to that object. The agent was capable to navigate correctly after learning.

The final behavior of the agent after learning is explained next. When observing an image, it starts from the *Null* state and then follows its learned policy until a motor action is selected. If in a state an object should be attended then, the salient region of the image (by applying the appropriate bias signals to the image) is transferred to the corresponding SVM classifier to be recognized. Then next state is determined and so on.

A hierarchical representation was formed in mind of the agent. This tree determines the order of attending to objects.

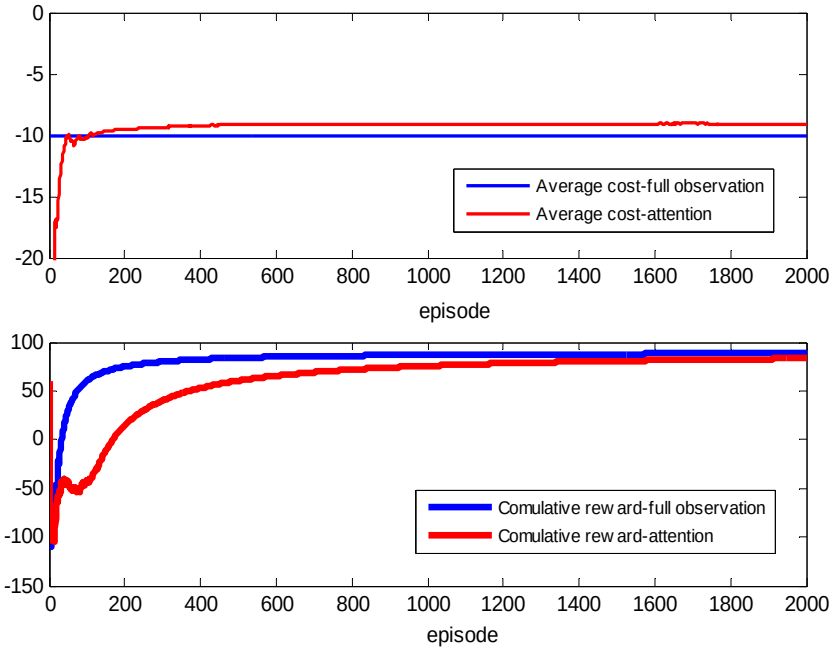


Figure 6.7 Top) Average cost, Bottom) Cumulative average reward during episodes. RL parameters: maxEpizodes = 2000, α = 0.7, γ = 0.9, ε = 0.9. Costs were defined as = [-4 -3 -2 -1] for objects [A B C D].

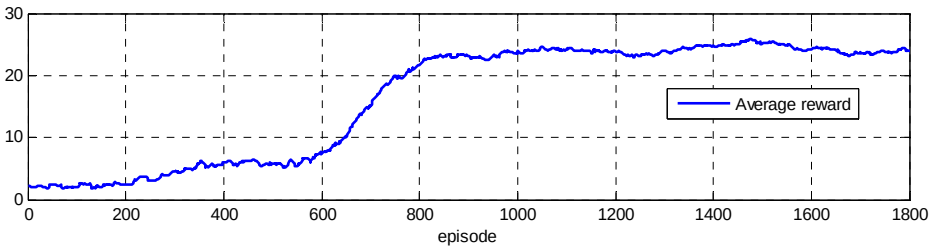
Motor actions unknown In this section, association of physical actions to scenes is not known in advance. Therefore, it has to discover both its representations (internal state space) and the best image-to–action mappings. The agent starts in the RL-fixed phase with a random position and a

random head direction in the map. It then receives an image randomly from 5 images associated with each state. To find its state, the agent traverses its attention tree down to a leaf node, afterwards it follows an ε-greedy strategy for action selection. Each episode is terminated when the agent reaches the goal state or goes off the road (hits the wall). Rewards and punishments are defined as in equation 14. Costs of attending to objects are not considered (zero cost).

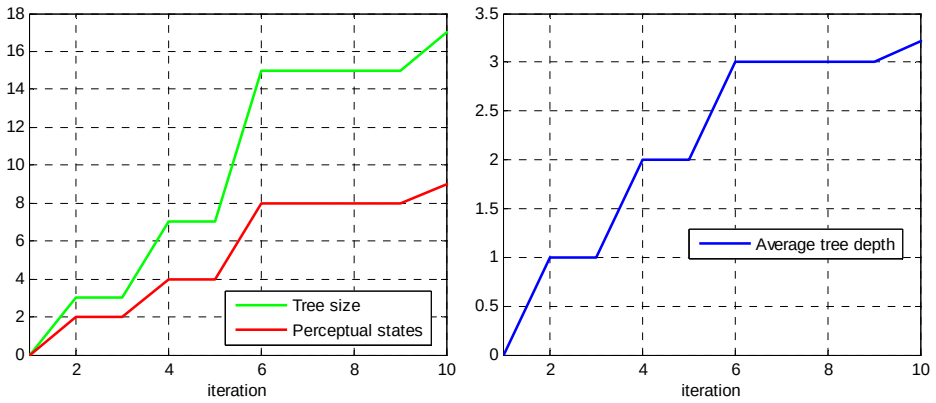
$$R = \begin{cases} +100: \text{ reaching the goal state} \\ -10: \text{ going forward and hitting a blocked way} \\ -5: \text{ turning and observing a blocked way in front} \\ 0: \text{ ordinary forward or turn} \end{cases} \tag{6.4}$$

Average reward is the sum of the rewards that the agent receives during an episode normalized by the length of that episode. Average tree depth is the average depth of all leaf nodes in the attention tree. Figure 6.8, shows the average reward of the agent, statistics of the generated tree like number of nodes and leaves and its average depth. It also shows error rate in the policy of the agent.

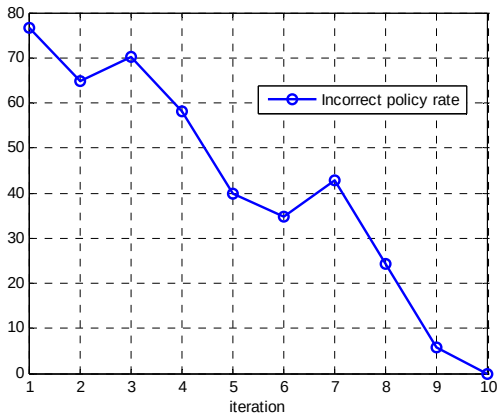
Final depth of the generated tree is 3.22, which means that the agent could do the task perfectly by attending to 3.22 objects in average. This value is below 4 objects in full attention case and shows 20% improvement in reducing the processing costs. It could be seen that, when perceptual aliasing decreases, average reward increases. As figure 6.8.b shows, average depth of the tree increases during learning. Figure 6.8.c shows the percentage of incorrect policy. After 10 iterations, it converges to zero that means that the agent has learned the optimal image to action mapping and there are no further aliasings in the tree. Final generated tree is shown in figure 6.9. Nine leaves (states) were generated for handling the task which is smaller than 12 states of the environment. Since some states have the same best actions, they were clustered under the same leaf nodes (states with plus signs in figure 6.9).



a)



b)



c)

Figure 6.8 a) Smoothed average reward in a window of 200 episodes, b) left: Number of nodes and leaves of the learned attention tree, right: Average tree depth, c) percentage of incorrect best action. Parameters: maxEpisodes = 200, ε = 0.8, memThreshold: 20, α = 0.9, γ = 0.9, aliasingThreshold=11. (For the map of figure 6.3, without pruning)

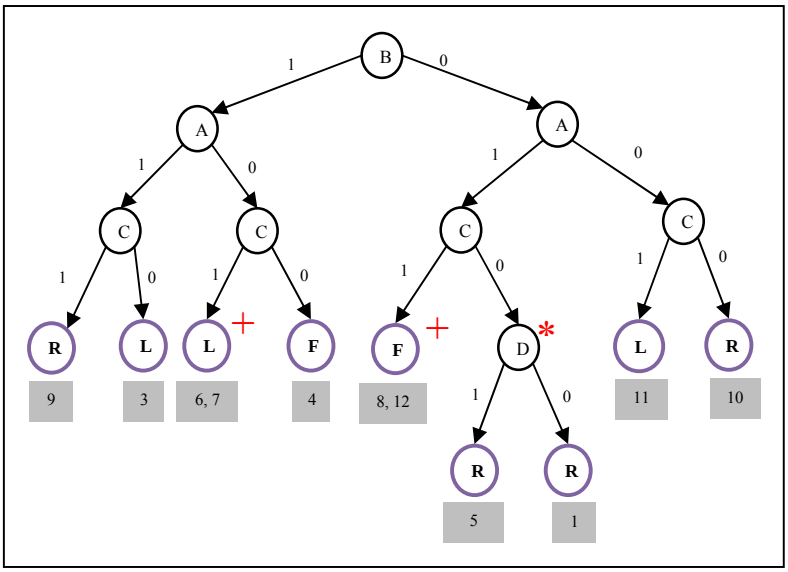


Figure 6.9 Learned attention tree for the map of fig 8 without pruning. Algorithm managed to generate a tree with 9 states. Letters inside the blue circles (leaves) are the learned motor actions. This tree resulted in 100% correct policy but is not efficient since the node marked with * has two children with the same best actions. Different states with the same best actions are clustered under the same leaves.

It could be verified from the tree in figure 6.9 that the algorithm generates inefficient trees. In its general form, it creates some unnecessary leaf nodes. For example, the node labeled with * in this figure has two children with the same best actions. Clearly, such nodes have anomaly and could be merged and be replaced with their parent node. Some nodes also do not absorb any experiences indicating that they are not necessary. To generate efficient trees, generated trees are pruned in some occasions, for instance after d iterations (here after the last operation).

Generated tree after pruning is shown in figure 6.10. Algorithm in this case succeeded to generate a more compact tree by clustering more states with the same best actions under the same leaf nodes. Final generated tree again achieved 100% correct policy rate with 6 states and average depth of 2.66 which are smaller than 3.22 and 9, average depth and number of states of the un-pruned tree in figure 6.9.

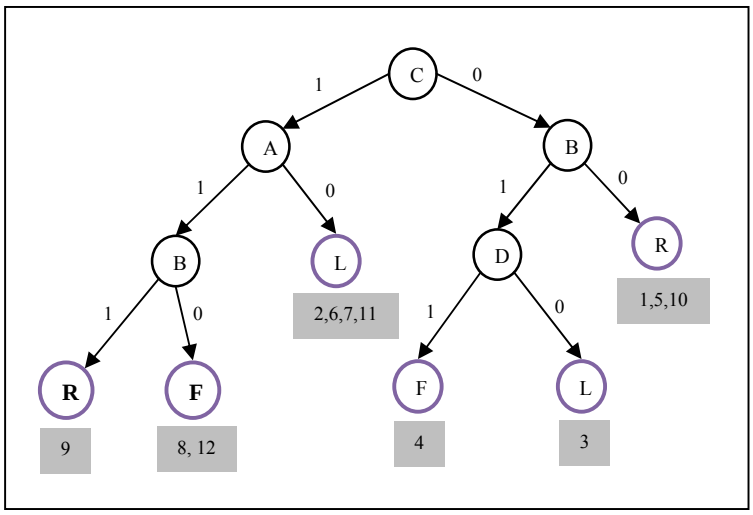


Figure 6.10 Learned attention tree for the map in fig 8 with pruning. Twelve states were clustered into 6 leaves. Algorithm succeeded to achieve the 100% correct policy in this case. Parameters: maxEpisodes = 200: $\varepsilon = 0.8$, memThreshold= 100, $\alpha = 0.9$, $\gamma = 0.9$, aliasingThreshold= 10.

We also applied our method to a more complex navigation environment. In this task, the agent moves between 11 positions of the map in figure 6.11. Neither motor actions nor visual representations are known in advance. The state space is of size $11 \times 4 = 44$. Actions and reinforcement signals are as in previous section. The agent has to connect an input image to the appropriate reaction without explicitly knowing its geographical localization.

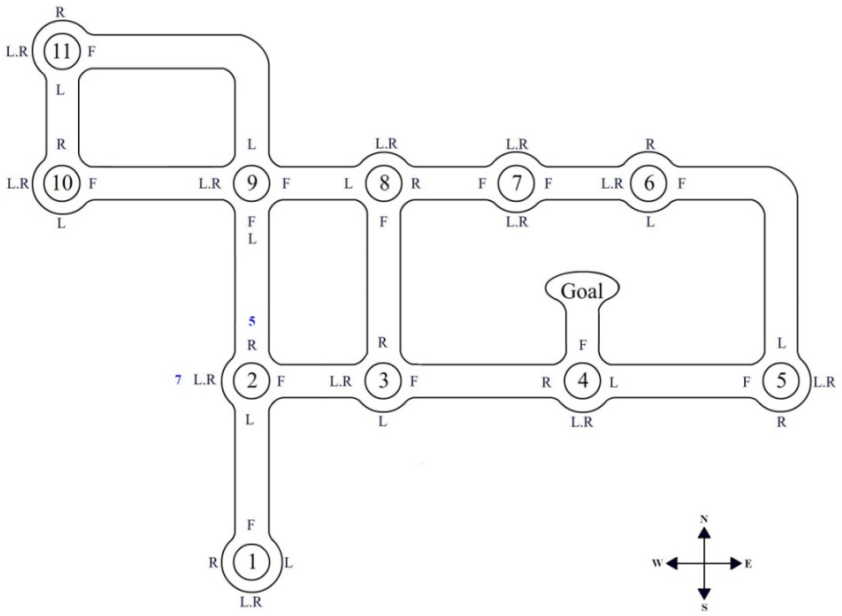


Figure 6.11 Complex navigation map. A subset of 5 objects is randomly embedded in each scene. Best actions are shown besides each state. In some states two actions are optimal.

Five scenes are associated with each state and there is no error in detection and recognition. Objects in the scenes (4 traffic signs plus the coke object) are randomly speared in 2D space and are not bound to specific spatial locations. A random assignment of the objects to the states is shown in table 4. Note that, the agent observes some same scenes in different states. Since there are 44 states in the map, and 31 combinations are possible with 5 objects (minus empty scene), therefore aliasing occurs in some states (the agent observes the same scene in different states with the same best actions). To avoid such aliasings, no same scenes were assigned to states with different optimal actions.

Table 6.2 A random assignment of objects to states of the map in figure 6.11.

1	**{B}**	**9**	**{A, B, D}**	**17**	**{A, C, E}**	**25**	**{A, B, C, E}**	**33**	**{A, B, D}**	**41**	**{A, B, C, D}**
2	**{A, B}**	**10**	**{A, B, C, E}**	**18**	**{B, C, D, E}**	**26**	**{D, E}**	**34**	**{A, D, E}**	**42**	**{A, C, D}**
3	**{A, C, D}**	**11**	**{A, C, E}**	**19**	**{D}**	**27**	**{A, D, E}**	**35**	**{D, E}**	**43**	**{A, C, E}**
4	**{A, C}**	**12**	**{B, C, D}**	**20**	**{A, D, E}**	**28**	**{B, D, E}**	**36**	**{B, D, E}**	**44**	**{B}**
5	**{A, B, C, D}**	**13**	**{B, D}**	**21**	**{A, B, D}**	**29**	**{A, D, E}**	**37**	**{A, B, C, D}**		
6	**{A}**	**14**	**{A, E}**	**22**	**{A}**	**30**	**{A, D}**	**38**	**{A, C, D}**		
7	**{A, C}**	**15**	**{A, D, E}**	**23**	**{A, C, E}**	**31**	**{B, C, D, E}**	**39**	**{A, C, E}**		
8	**{B, D, E}**	**16**	**{A, D}**	**24**	**{B, D}**	**32**	**{A, B, D}**	**40**	**{B, C, D}**		

Generated tree for this experiment without pruning and with deterministic observation (without uncertainty) has 23 leaves. Algorithm converged to 100% correct policy rate after 55 iterations (phases). Final tree has the average depth of 4.2 which is smaller than 5 objects (full attention). So, the agent attends to 4.2 objects in average while having the same performance. However, again this tree is not efficient since there are some nodes with no patterns classified

under them. In some situations, leaves below a node have the same best actions which evidentially are not necessary and could be merged in their parent node.

Figure 6.12, shows the average reward of the agent, statistics of the generated tree like number of nodes and leaves and its average depth without pruning. It also shows error rate in the policy of the agent. Figure 6.13 shows the generated tree after pruning. Algorithm generated 7 states with average depth of 3. Therefore pruning could extensively help optimizing the trees.

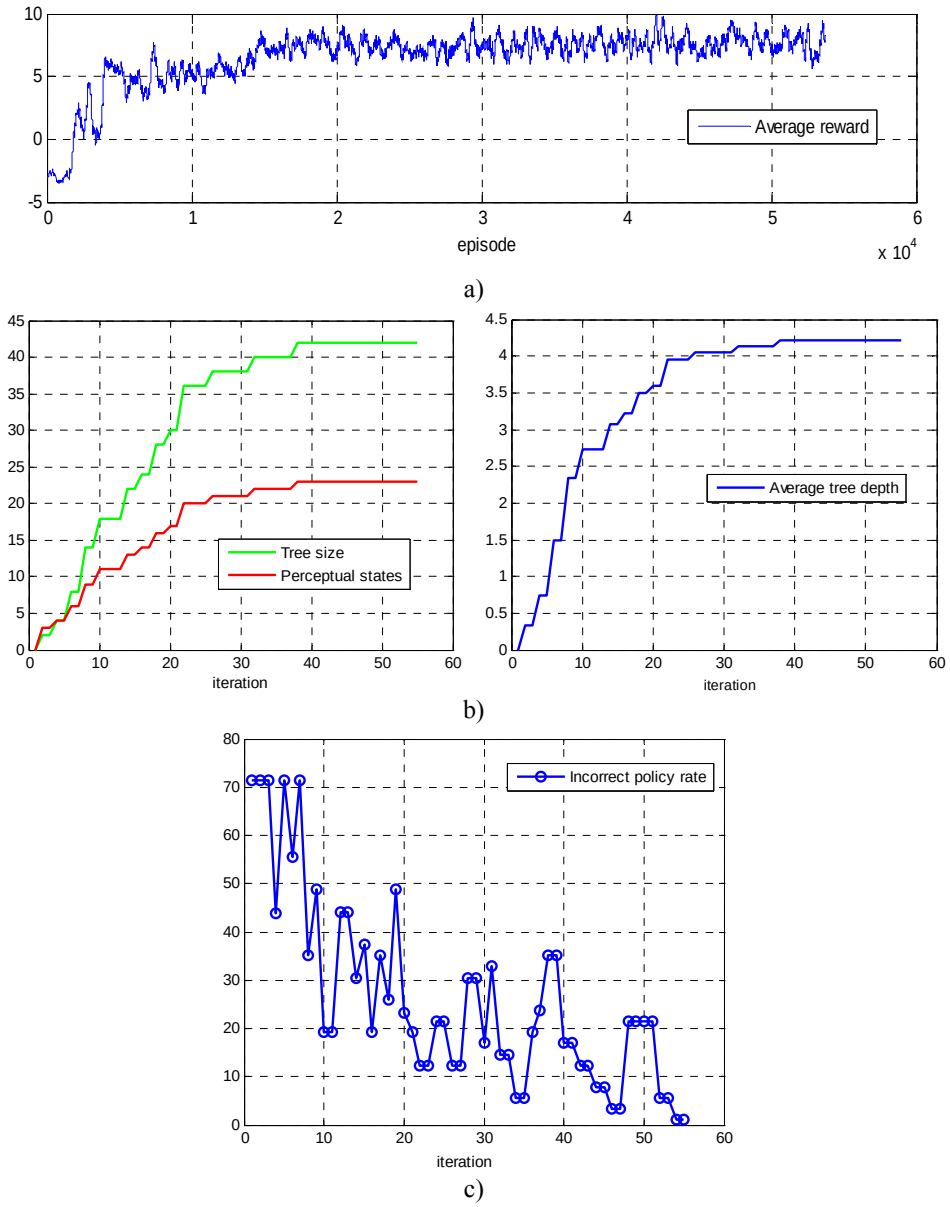


Figure 6.12 a) Smoothed average reward in a window of 200 episodes, b) left: Number of nodes and leaves of the learned attention tree, right: Average tree depth, c) percentage of incorrect best actions. Parameters: maxEpisodes = 200, ε = 0.8, memThreshold: 700, α = 0.9, γ = 0.9, aliasingThreshold=120.

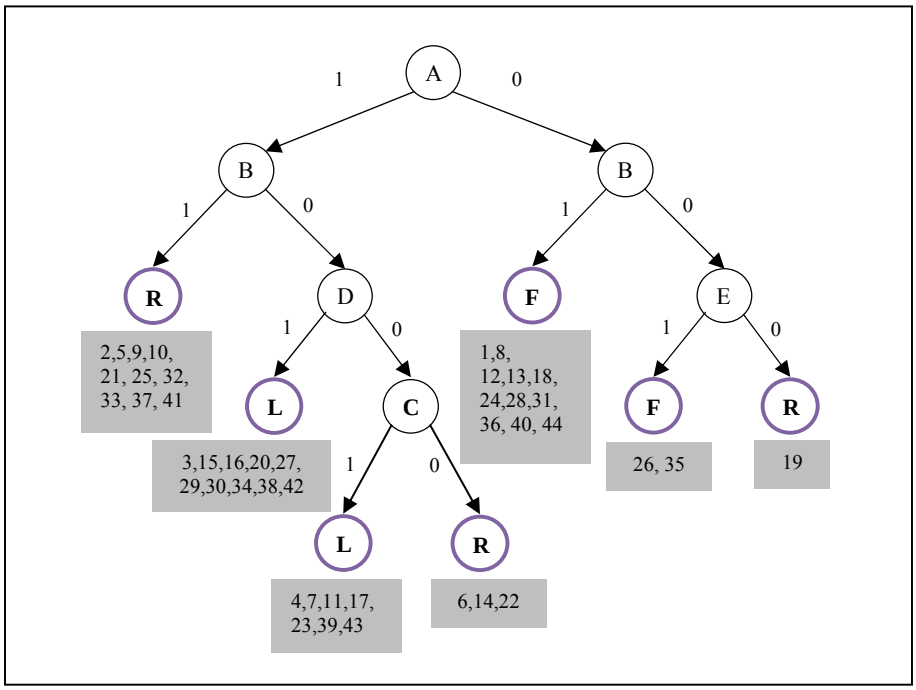


Figure 6.13 Learned attention tree for the map of fig 16 after pruning. Forty four states were clustered into 7 leaves. 100% correct policy was achieved.

6.4.2 Uncertainty analysis

An important aspect of decision making in real-world situations is the fact that an intelligent system should be able to deal with uncertainty from numerous sources. For instance, a major source of uncertainty for a robotic agent is its sensors, which are often noisy and have only a limited view of the environment. Since both saliency model and the HMAX have uncertainties, this problem also applies to our model. In this section, we analyze how uncertainty in the perception of the agent affects his behavior over the map of the figure 6.3. Each observation of the agent is incorrect by probability P_u. When the agent traverses its attention tree and has to attend to an object, it might get an incorrect result with probability P_u. If P_u=0, the agent's observations are always correct. Therefore behavior of the agent is deterministic and it always does the same action (policy) for the same observation. However when observations of the agent are noisy, then the agent develops a probabilistic action selection strategy. Since no formulation is involved, the only way for the agent to discover such probabilities is by maximizing its reward. As figure 6.14 shows, low magnitude noise does not degrade the behavior of the agent a lot (for $P_u = 0$ and $P_u = 0.03$). However when the noise percentage is increased to 0.1 or 0.4, algorithm does not converge to a near optimal policy. Increasing the uncertainty resulted in gaining less average reward and more faulty policies.

Therefore, the agent could compensate low magnitude of uncertainty in the saliency model and HMAX by optimizing its action selection strategy.

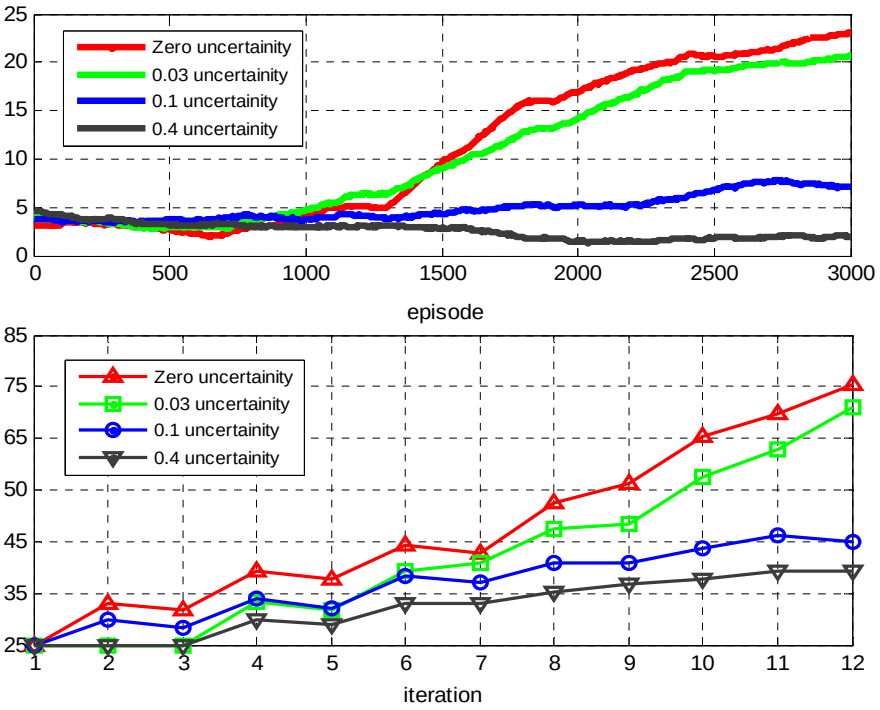


Figure 6.14 Top: Cumulative average reward of the agent for different noise levels. Bottom: Cumulative percentage of correct policy during learning (For the map of figure 6.3). Results are averaged over 7 runs. Analysis was done using pruning and the same parameters as in figure 6.8.

6.4.3 Comparison with decision trees learned with C4.5 algorithm

The best advantage of interactive learning of action-based representations and top-down visual attentions, through discretizing state space when aliasing occurs, is that such learned representations are optimal for the behavior of the agent. In this section we compare the tree learned with our algorithm for learning object-based visual attention with the tree generated offline by the basic C4.5 algorithm (Quinlan, 1993) over the map of figure 6.11. In C4.5, an object is selected for a node which maximizes the information gain according to equation 15.

$$o^* = \underset{o}{argmax}\ (Gain(y, o)) \tag{6.5}$$
$$= \underset{o}{argmax}(Entropy\{y\}\left(\frac{|L_+|}{|L|}Entropy\ \{y_+\} + \frac{|L_-|}{|L|}Entropy\ \{y_-\}\right))$$

In above formula, o* is the selected object, o is the object which is being checked. y is the set of all items in a node, y_+ is the set of items with object o and y_- is the items without o. |L|, $|L_+|$ and $|L_-|$ are the size of these sets respectively.

For states in the map of figure 6.11, with two best actions, one was selected in random. Leaned tree with the C4.5 algorithm is shown in figure 6.15.

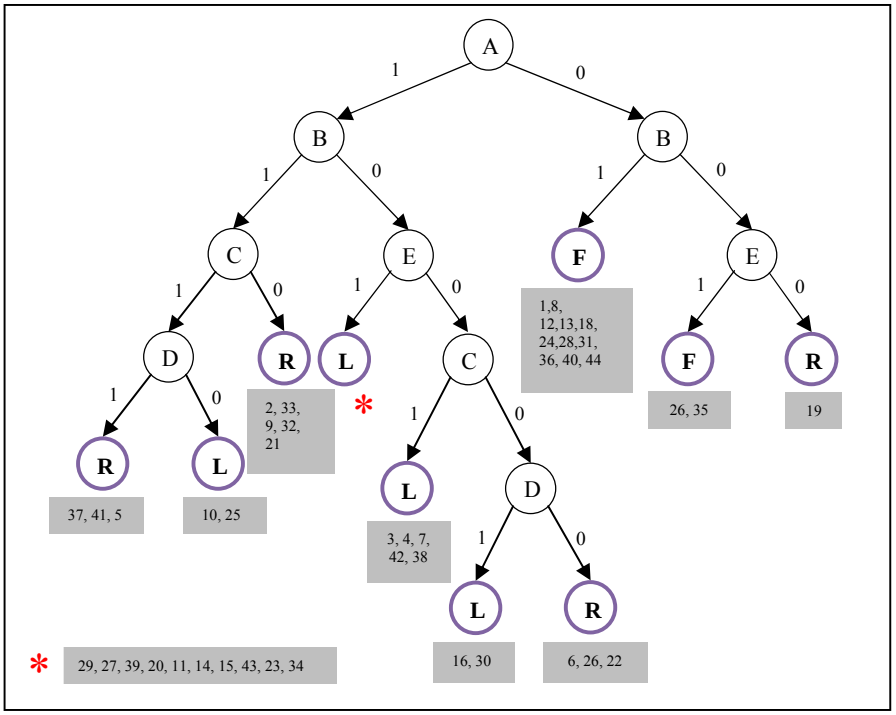


Figure 6.15 Offline learned decision tree with C4.5 algorithm. Average depth of the tree is 3.6 and the correct policy using this tree over the map in fig 16 is 100%.

C4.5 managed to generate a tree with 10 perceptual states (leaves) with the average depth of 3.6. Percentage of the correct policy using this tree is 100%. This shows that while C4.5 algorithm, achieved to find an optimal policy, it generated a suboptimal representation.

6.5 Discussions

In our proposed method, state space is incrementally constructed from a single *Null* state. In order to maximize its cumulative reward, the agent has to continuously create new states and learn the best physical actions associated with them. Our method for learning task-based visual attention discretizes state space when perceptual aliasing occurs. Aliasing is measured using TD errors (Bellman residuals). In RLVC (Jodogne & Piater, 2007), SIFT features are used for building a representation tree. In each internal node of this tree the existence of a specific SIFT feature is examined. Learned tree by RLVC is loosely related to visual attention because it always calculates the SIFT features over the entire scene. This exhaustive search for a SIFT feature brings advantages like partial invariancy to orientation, scaling and translation, but has the drawback of high computational complexity. It might be advantageous to extract SIFT features at some specific locations (eg. at the focus of attention) to reduce the computational complexity of this approach. However remedies should be anticipated to solve the sensitivity of this approach to image transformations. In our object-based method only a region of the image is searched for checking the existence of an object. This region is selected via the saliency-based model of visual attention which is biased toward specific objects. Therefore our method eliminates this shortcoming of RLVC.

Our results show that the quasi-static learning of actions and attentions over saliency and HMAX models could solve complex navigation problems. Instead of detecting and recognizing all the objects in a scene, or processing entire image content, the best action associated with that scene could be found by only attending to a few objects. For example in the complex map in section 4.1.2, 3 numbers of attentions and object recognitions were enough for action-based classification of the scenes, instead of recognition of 5 objects. Considering that attention windows and scenes were of size 50×50 and 360×270 pixels respectively, on average 0.07 percent of pixels were analyzed with average 3 saccades. This clearly shows that object-based attention could largely save computational resources. Also it was shown that pruning could significantly enhance the generated attention trees. Such trees, while being compact, result in optimal policies. Generated trees had smaller number of leaves (states) and average tree depths. This means that the agent could attend to a smaller number of objects in average to solve a task. It was also shown that training RL with noisy data could compensate low-magnitude noises, but larger values of noise significantly degrade the RL convergence. Generated trees with our model are more efficient than offline learned trees with the C4.5 algorithm in terms of average tree depth. This clearly shows the importance and benefit of action-based and attentive representations over hand designed and offline solutions.

Since online learning of weights of the saliency model is not reasonable due to degrading the RL convergence, here we divided learning top-down object-based attention into two parts. First saliency model is biased offline and efficiently (in terms of reducing the cost of bottom-up attention) toward objects of interest for the agent. These objects are those the agent might observe when doing a specific task online. Then RL is used to discover when and how these learned signals should be applied interactively. Actually our method proposes a basic framework to use the bottom-up saliency model, standard model of object recognition and RL techniques for top-down attention learning. To solve the remaining problems, solutions from RL approaches can be borrowed. For instance, when aliasing occurs, a short term memory component could be used to omit the aliasing by remembering previous actions, attentions and observations.

When gathering memory under a leaf node, it is assumed that the agent knows which objects are in the scene. Actually this assumption limits the work since it is not easy to find all the objects in a natural scene. To the best of our knowledge, an efficient solution for this problem does not exists yet. It will be helpful to start working with large databases of natural scenes containing segmented and tagged objects (like *LabelMe* dataset (Russell et al., 2008)).

Other efficient and task-specific object detection and recognition methods could be substituted with model components. For example for environments when the agent is concerned with specific class of objects (like faces, cars, etc.) solutions for detection of such objects might lead to more efficient top-down solutions. Clearly such case-based object detection solutions are more efficient than a general solution which is assumed to detect any object. For example there are successful face detection approaches in the literature. Same thing also applies for object recognition. This is mainly because of this shortcoming of the saliency model which does not consider the structure of the objects in searching. (A learning mechanism for search will help). A recent attempt is done in (Walther & Koch, 2006) by using the idea of proto-objects to help the search mechanism. Another study has proposed a Bayesian formulation in order to consider the context and the backgrounds that objects might appear in (Torralba, 2003). However, it seems

more studies should be done experimentally and computationally to discover the role of object structure in visual search tasks.

From the point of view of robotics, if a method, based on low-level features (like RLVC with SIFTs) enables an agent to solve its task adequately and efficiently (low computation), then it is sufficient. Now this question arises: "If a task could be solved with low-level features, what is the benefits of working with global representations like objects?". There are two reasonable answers to this question. First, such methods while proposing solutions in robotics are in accordance with some previous findings on visual attention like bottom-up and top-down attentional mechanisms (object-based and spatial). They might also be useful for understanding how high level top-down attention is mediated by different parts in the brain when an agent has to interact with its environment and continuously make decisions. Decision making studies have previously used RL approaches to understand decision making processes in the brain but, there are not many computational studies trying to explain experimental data of attentional behaviors (in conjunction with decision makings) using reinforcement learning. As we mentioned in section two, some attempts have been made for modeling eye movements with RL which to some extent explains cortical attentional mechanisms. Second reason, why a simple solution which uses low-level localized visual features while applicable might not be appropriate, is because an autonomous agent in order to communicate and interact with its surrounding environments has to do a repertoire of tasks. Low-level features might be the best when an agent has to do a limited, well-defined single task. Actually the reason why humans deal with objects instead of low-level features might be because they have to perform various cognitive tasks. This way, objects help the agent to make abstractions in its world which leads to generalizations in its behavior. There are also evidences claiming that humans manipulate objects in order to understand a scene, instead of dealing with low-level features.

6.6 Summary and conclusions

In this research, we proposed and partially implemented an overall biologically inspired model for top-down object-based visual attention control. In our model, we considered how task demands, actions and the bottom-up cues influence attention. Our results support the idea that the nature of the bottom-up attention is low-level mechanisms, while top-down attention is more like a control or a decision making problem. To solve such problems optimization approaches have been followed in engineering literature.

Rather than scanning the image from top-left to bottom-right, to detect an object in the scene, or using global representations (which usually need many computations), our model just looks at a small number of spatial locations. The information about what and where to look comes from the top-down knowledge learned by RL during learning a task. Links of model components to biology were discussed in each section of the paper.

Our main contributions were proposing a method to find the low-cost weights of the saliency model to bias it for object detection and a top-down mechanism for controlling the bottom-up saliency model for doing a task.

Our approach is in accordance with the important biological evidence that brain has adopted a need-based approach for visual representations. This means that representations and other cognitive capabilities of the agent are altered or refined in order to optimize its behavior and interactions with the world it lives in. A minimal solution (representation) is enough and no

further detail (over-completeness) is needed. This mechanism is well captured by our model for deriving visual representations and top-down task-based attentions. Actually representations and attentions are optimized in our model by maximizing cumulative reward of the agent (behavior-based approach). For gaining maximum reward those states with aliasing are expanded into other nodes which then lead to more refiner representations.

Major next step is to extend the work for continuous interactive visual environments where the agent continuously receives visual information from the environment and has to perform continuous and complex motor actions.

Integration of bottom-up and top-down attentional mechanisms has not been studied and shown in previous computational models which leaves places for future works. For example, a computer operator has learned how and where to move his eyes when working with a computer program from his past experience. This shows the task-driven component of visual attention. When suddenly he closes a screen which needs something to be saved, suddenly a pop-up message box appears and alarms for saving. Our top-down model could be tried to explain the top-down component of this sample task using eye movement data. However, how the bottom-up part should be integrated in the model needs further investigation.

In this paper, we only considered the perceptual aliasings mediated by the insufficient observations. If the agent could observe all the necessary data, therefore no aliasing occurs. Some aliasings are because of the lack of knowledge of the agent of its previous actions and observations. For example when two same scenes, associated with two states with different best actions, are presented to the agent, then it could not discover its optimal policy because of the contradiction in decision making. Such problems are remedied in RL by equipping the agent with a short-term memory that keeps track of the previous actions, attentions and observations of the agent to satisfy the Markov property. This idea could be used to extend our model for performing tasks with this aliasings.

We only considered the costs of attending when biasing the saliency model. Costs of objects could also be considered for deriving trees with less average cost of attending. This way, for example when evaluating an object for expanding a node, its value should be normalized with its cost. That might result in choosing low cost and high discriminant objects at higher levels (smaller depths) of attention trees.

It would be interesting to apply the ideas of dynamic and interactive state space discretization to behavioral and neurophysiological studies (especially on attention) and analyze their applicability for explaining some experimental data. The importance of this topic is better understood when we note that previous studies have mainly been focused on simple laboratory experiments. That's because such experiments become complicated when a subject should interact with its surrounding environment.

Chapter 7
Saliency Maps for Attentive and Action-based Scene Classification

Ali Borji, Majid N. Ahmadabadi, Babak N. Araabi

To be submitted.

Abstract. This work proposes an approach for attentive scene classification by extracting and matching visual features only at the focuses of visual attention instead of the entire scene. Analysis over a database of natural scenes demonstrates that regions proposed by the saliency-based model of visual attention are robust to image transformations. Using a nearest neighbor classifier and a distance measure defined over the salient regions, we obtained 97.35% and 78.28% classification rates with SIFT and C2 features at five salient regions of size 140. Classification with features extracted from the entire image results in 99.3% and 82.32% using SIFT and C2 features, respectively. Comparing attentional and adhoc approaches shows that classification rate of the first is 0.95 of the second while it needs feature extraction at only 0.31 of spatial regions. An approach for simultaneous learning of physical actions and representations known as RLVC is also extended by limiting SIFT search and extraction to only salient regions. Results prove that RLVC has still the same performance with a huge decrease in computational complexity. Overall, our results prove that efficient scene classification, in terms of reducing the complexity of feature extraction is possible without a significant drop in performance.

7.1 Introduction

A large number of approaches for scene classification have been developed which can be classified into the following three categories. 1) *Low-level feature based schemes:* which represent scenes by global visual information (Boutell et al., 2002), including color, texture, and shape have been successfully utilized in indoor/outdoor, city/landscape and forest/mountain applications. 2) *Local feature based schemes:* represent scene images with detected interest points (Harris and M. Stephens, 1988, Lowe, 2004) (or regions) based on some descriptors (Daugman, 1980, Lowe, 2004). Local-global features (Lazebnik et al., 2006) based schemes utilize both the global spatial information and the local descriptors of interest points (or regions) to represent scene images to achieve a robust classification. 3) *Biologically inspired feature based schemes:* classify scenes by mimicking the process of visual cortex in recognition tasks. Recent reports from both neuroscience and computer vision have demonstrated that biologically plausible

features (Oliva & Torralba, 2001, Siagian & Itti, 2007, Serre et al., 2007) are attractive in visual recognition.

Our work in this paper falls in the third category and aims to apply the inherent capability of human brain known as attention to outdoor scene classification. Visual attention is the capability of biological vision systems by which humans and animals select most salient regions of a scene to later concentrate higher vision tasks on those areas. This explains to some extent high efficiency of humans in every day detection and recognition tasks. Saliency means that a subset of image regions conveys more information for a biological creature, to optimize his behavior. It can be determined by bottom-up, image-based characteristics which are mainly derived by early visual areas like V1 and V2 and also top-down task-driven cues from higher areas like LIP, V4 and PFC.

From a theoretical perspective, attention is a solution for the problem of information overloading and computational complexity. This problem has attracted many concerns in computer vision and cognitive robotics where large amount of data has to be processed efficiently to guarantee a limited response time. It is also of interest to produce human-like behavior in human-computer interaction applications.

While a large body of research has been focused to understand biological underpinnings of visual attention and its modeling, it has been only until now that its applications have begun to emerge in computer vision and robotics. It has also been incorporated to solve complex visual tasks like 3D reconstruction (Pollefeys, et al., 2004), robot navigation and localization (Lowe & Little, 2002, Frintrop & Jensfelt, 2008), scene (Siagian & Itti, 2007), object and face recognition (Lowe, 2004).

Several computational implementations of models of visual attention have been published (Frintrop et al., ACM 2009) in last decade. However, little work has been done in investigating their benefits for scene and object classification in natural environments (Walther & Koch 2006). In this paper, applicability of saliency-based model visual attention is examined for scene classification. We first investigate whether and to what extent images contain similar visual content at salient regions or are repeatable. We furthermore aim to demonstrate how this information could be utilized for outdoor scene recognition.

Repeatability of a different class of saliency, discriminative saliency, defined by Gao et al. have been studied in Gao et al. (CVPR 2007). They speculate that saliency serves to maximize recognition. Using SVM classification, they have shown that discriminate saliency detector (DSD) leads to higher recognition rate than Scale Saliency Detector (SSD) (Kadir & Brady 2001) and Harris Saliency Detector (HSD) over face, motorbike and airplane datasets. They have also compared the repeatability of DSD with Hessian-Laplace (HesLap) (Mikolajczyk & Schmid, 2004) and Maximally Stable Extermal (MSER) (Matas et al., 2004) detectors. There are some concerns about the idea that recognition is the major goal of saliency. Some biological findings of attention contradict this claim. Saliency for recognition means that some feedback mechanisms for example in form of projections from higher brain areas modulate the early areas. This is despite the facts that bottom-up attention in visual cortex is feed-forward, very fast (80 ms) and only based on raw mage characteristics. Furthermore there is no evidence whether this concept of saliency matches with human eye movement behaviors or not, while we have evidences that bottom-up saliency (Itti et al., 1998) generates scan- paths correlated with saccadic eye movements (Ourehani et al., 204, Electronic Letters).

Local image descriptors have become a very powerful representation of images in categorization and recognition tasks (Schmid & Mohr, 1997, Lowe, 2004). Much of their success is due to their distinctiveness and to the fact, that this type of image representation is robust to occlusions and affine transformations. While in some applications like 3D reconstruction, a large number of local features is needed in some others only a few suffices. In many real-world applications where computational resources are limited, a small number of descriptors are preferred because it greatly reduces the complexity of matching algorithms. In (Walther et al. 2005), Walther et al. have proposed a method for multiple object recognition in cluttered scenes using SIFT features. First a salient region is detected using the basic saliency model and then a patch is grown to fit an object extent. SIFT feature extraction is done at this patch for matching the object at to the learned objects. Similar to this work, we also extract SIFT features at some salient spatial region but for scene classification.

From another perspective the proposed algorithm reduces the computational complexity of SIFT feature extraction algorithm. SIFT algorithm generates a large number of features e.g about 2000 for a 500×500 image (ms per image on average). There have been some attempts to speed up this process by modifying the way they are calculated. We claim that it is possible to extract SIFT features at a subset of spatial locations without a major drop in performance. Instead of feature extraction over the whole image, features are extracted over some moderately stable salient regions and are then matched.

Our scene classification algorithm consists of three integrated modules 1) a saliency detector for selecting a subset of scene regions 2) a matching algorithm which matches image content at the salient regions and returns a distance value between images and 3) a classifier for class label prediction. All these stages to some extent cope with biological findings.

In contrast with many machine vision applications, in robotics processing resources are limited and small response times has to be guaranteed. This makes it necessary to design efficient solutions in terms of accuracy and speed for such problems. Human brain as a physical system faces the same problem where huge amount of sensory information is received but only a very small amount of them has to be processed. Biology has designed a solution known as visual attention where some image regions are selected in a task-relevant fashion in order to be processed in detail. In this way, attention ants as filter which allows only informative and task-relevant areas to pass to higher visual brain areas.

Several researches have been already proposed to model visual attention and benefit from them in machine vision and robotics context. Visual attention could act by selecting image image regions or objects in both bottom-up and top-down manners. In bottom-up attention, image regions are selected only based on physical image characteristics like luminance contrast, color contrast, orientation contrast, etc in an open loop manner. In top-down attention selection criteria dynamically changes based on task demands, motivations, emotions, etc. While there has been lots of efforts to model bottom-up component of visual attention like saliency model (Itti et al., 1998), much interest has been raised to understand and model top-down component of visual attention due to its applicability in many computer vision and robotic tasks. A line of study in machine learning known state-space discretization deals with learning top-down attention indirectly and in a different fashion from other top-down modeling studies. In these studies, representations (states) of the agent are dynamically learned in an interactive fashion (with agent acting in the environment) whenever perceptual aliasing. In this way, decisions and actions are

learned simultaneously. The structure coding for representation allows directing attentions too. This has been generally formulated first in (McCallum, 1995) and has been extended to visual domain by Jodogne and Piater (2007).

RLVC consists of two interleaved learning processes: 1) an RL unit which learns image to action mappings and 2) a binary image classifier which incrementally learns to distinguish visual classes when perceptual aliasing occurs. The classifier acts like an attention tree by checking whether a specific SIFT feature is present in the image or not. The main drawback with the RLVC is its exhaustive search over entire image for checking the existence of a specific SIFT feature which could hardly be considered as an attention method since it does not select a subset of image regions. In our method, introduced in the second part, like RLVC, an action-based decision tree is constructed. We use the basic U-TREE algorithm for refining the internal representations of the agent. However, our method solves this shortcoming of RLVC, exhaustive search for a specific feature, by extraction and matching visual features only at the focuses of attention proposed by the bottom-up saliency-based visual attention model.

In top-down attention, focuses of attention (saccade locations) has to be learned. Saliency could help efficient scene classification and action-based scene classification when actions and attention has to be learned in order to perform a real-time task. Two alternative solutions for above problems are 1) to learn to process scene sequentially for example by processing regions proposed by saliency model like saccadic eye movements (Or to constrain resources to locations to a meaningful and biologically-inspired approach like saliency model) or 2) Process all salient regions in parallel and constrain feature extraction to these moderately stable regions. The first approach is subject of our other studies (Borji et al., 2009) but in this paper, we follow the second approach for scene classification for situation when the goal is to maximize correct classification rate and when the goal is to maximize reward or correct policy rate.

In order to select some salient regions we need a saliency detection approach which has these three characteristics 1) be computationally inexpensive and fast 2) produce stable regions through image transformations and 3) be a suitable in both biology and engineering in sense of coping with biology, computation complexity, applicability , etc. Since saliency model (Itti et al., 1998) satisfies these conditions, it is used for saliency detection in this paper.

The rest of this chapter is organized as follows. In section 7.2, saliency-based model of visual attention is briefly explained. Attentional scene classification using saliency concept is presented in section 7.3. Section 7.4 extends the RLVC by constraining feature extraction to salient regions.

7.2 Saliency-based model of visual attention

Several studies in computer vision have previously investigated the paradigm of visual attention and now there is an accepted model known as the basic saliency-based model (Itti et al., 1998) which copes with biological concepts known to some extent. This model generates a 2-dimensional topographical map that encodes stimulus conspicuity or saliency at every scene location. Saliency map is constructed as follows.

The input image I is sub-sampled into a Gaussian pyramid (Burt & Adelson, 1983), and each pyramid level is decomposed into channels for red (R), green (G), blue (B), yellow (Y), intensity (I), and local orientation (O_θ), then $R = r - (g + b)/2$, $G = g - (r + b)/2$, $B = b - (r + g)/2$, and $Y = r + g - 2(|r - g| + b)$ (negative values are set to sero). Local orientations (O_θ) are obtained by applying steerable filters to the images in the intensity pyramid I (Burt &

Adelson, 1983). From these channels, center surround "feature maps" are constructed and normalized:

$$\begin{aligned}\mathcal{F}_{I,c,s} &= \mathcal{N}(|\mathcal{M}_I(c) \ominus \mathcal{M}_I(s)|), I = \{I\} \\ \mathcal{F}_{C,c,s} &= \mathcal{N}(|\mathcal{M}_C(c) \ominus \mathcal{M}_C(s)|), C = \{RG, BY \\ \mathcal{F}_{O,c,s} &= \mathcal{N}(|\mathcal{M}_O(c) \ominus \mathcal{M}_O(s)|), O = \{0°, 45°, 90°, 135°\}\end{aligned} \tag{7.1}$$

where $\ominus$ denotes the across-scale difference between two maps at the center (c) and the surround (s) levels of the respective feature pyramids. $\mathcal{N}(.)$ is an iterative normalization operator (see (Itti & Koch, 2001) for details). The feature maps are summed over the center-surround combinations using across-scale addition (+) and the sums are normalized again:

$$\mathcal{F}_l = \mathcal{N}\left(\sum_{c=2}^{4}\sum_{s=c+2}^{c+4}\mathcal{F}_{l,c,s}\right), l \epsilon \{I, C, O\} \tag{7.2}$$

For the general features color and orientation, the contributions of the features dimensions are linearly summed and normalized once more to yield "conspicuity" maps. For

$$C_I = \mathcal{F}_I, C_C = \mathcal{N}\left(\sum_{l\epsilon C}\mathcal{F}_l\right), C_O = \mathcal{N}\left(\sum_{l\epsilon O}\mathcal{F}_l\right) \tag{7.3}$$

All conspicuity maps are combined into once saliency map: $S = \frac{1}{3}\sum_{k\epsilon\{I,C,O\}} C_k$.

Maximums of the final saliency map are the focuses of attention (b in our experiments). Despite the main model where maximums are sequentially detected using a WTA to generate a serial attention behavior, we use a simple max operation to speed up the process and generate a parallel attention mechanism. For a 640×480 images, it takes 50 ms to extract 5 salient points on average.

7.3 Attentive Scene Classification

Here is the aim is to construct a classifier for scene classification. Each scene is represented by a bag of features at either whole image or salient regions. Then a distance (similarity) measure is defined and then a classifier is built and is evaluated over a separate set of images. There is no interaction with the environment here and the only goal is to maximize correct classification rate.

7.3.1 Repeatability of Salient Regions

Here we explore to what degree salient points are repeated in different instances of the same scene class. To this purpose, we calculate the similarity of scene regions at the focuses of attention. This is indirectly in accordance to the repeatability criteria used to evaluate the quality of local feature detectors.

Features In our approach, a within fixation processing extracts features at focuses of attention for representing scenes. Two classes of features: SIFT (Lowe, 2004) and C2 features (Serre et al.,

2007 PAMI) are employed. The rationale behind selection of these features is comparing a local-appearance approach based on interest point and local descriptor concepts and another geometry-based approach to find out which feature type works well with saliency maps.

Sift features are distinctive features useful for reliable matching between different views of a scene or an object. They are invariant to image scaling and rotation, and partially invariant to changes in illumination and 3D camera viewpoint. SIFT is a staged approach, the first stage being the detection of salient key points. Key-point candidates are detected as local extrema of difference-of-Gaussian (DoG) filters in a Gaussian Scale Space of the input image. These candidates are then subpixel interpolated. Key points showing low contrast or lying on edges are discarded, for they are not stable. Invariance against scale and rotation is achieved by assigning a scale (according to the level of the scale space pyramid the key point was detected in) and an orientation (according to the principal orientation of gradients in a region around the key point) to each key point. Note that, in this step, points may be duplicated with different orientations if the local orientation histogram has multiple prominent peaks. Finally, the local image descriptor (i.e., the features) is calculated as a collection of smoothed histograms of gradient orientations and magnitudes over the local image region. The size of the feature vector depends on the number of histograms and the number of bins in each histogram. In Lowe's original implementation, a 4-by-4 patch of histograms with 8 bins each is used, yielding a 128-dimensional feature vector. We use a MATLAB/C implementation of the SIFT algorithm provided by Vedaldi (Vedaldi, 2006, TR).

C2 features are introduced by Serre et al. in (Serre et al., 2007). They have developed a computational model which based on the hierarchical organization of the visual ventral stream where shapes, object, face and scene recognition is performed. Simply in these models, low-level features which have high position and scale specificity are combined in a hierarchy to produce complex invariant features. This model starts with an image layer of gray-level pixels and builds simple (S) and complex (C) layers alternatively. Neurons in S layers convolve the image with a set of local filters to extract features and S units pool their inputs from previous layer to increase invariancy. (e.g. max operation). In this paper, we used the modified HMAX model (Mucth & Lowe, 2005) which learns a dictionary of prototypes patches in the C1 layer from a set of sample images. Number of learned patches determines the dimensionality of the learned feature vector. C2 feature provide rich structural information useful for recognition and matching and incorporating them with visual attention may give insights of interaction of visual attention and object recognition in human visual system (Walther & Koch, 2006).

Distance Measures Regarding the second step in scene classification, it is needed to define a distance or a similarity measure between two images to construct a classifier. We define distance measures for each type of above mentioned features. Given a set of m images $I = \{I_1, I_2, \dots, I_m\}$, a bag of features representation for the *i*th image with q SIFT features is:

$$F_i = \{f_{i1}, f_{i2}, \dots, f_{iq}\}, f_{ij} = (x_{ij}, y_{ij}) \tag{7.5}$$

where x_{ij} is the value of the *j*th SIFT feature (a 128-dimensional vector) and y_{ij} is the position of this feature in image coordinate frame.

The idea behind the state-of-the-art algorithms for matching and recognition is that they measure the similarities between *all* local features within compared images. We define a distance measure which is the average of the best matches for all features of the *i*th and *j*th image as:

$$D(F_i, F_j) = \frac{1}{2}\left(d(F_i, F_j) + d(F_j, F_i)\right) \tag{7.6}$$
$$d(F_u, F_v) = \frac{1}{n_u}\sum_{l=1}^{n_u} \min_{t=1..n_v} norm(f_{ul} - f_{vt})$$

where $norm$ is the Euclidean distance between two SIFT features. Above distance measure does not consider the fact that features remain at spatial vicinity of each other most of the time with image transformations. Following this constraint distance measure in (7.7) may enhance feature matching and hopefully recognition.

$$d(F_u, F_v) = \frac{1}{n_u}\sum_{l=1}^{n_u} \min_{t=1..n_v} norm(f_{ul} - f_{vt}).(H - e^G) \tag{7.7}$$
$$G = -\frac{1}{2\sigma^2}\left(norm(x_{uk} - x_{vp})^2\right)$$

H in (7) is a large positive number (= 1000) therefore $(H - e^G)$ is small when distance between features is small and hence results in small distance.

For images represented by C2 feature vectors, distance measure is simply the Euclidean norm of their difference:

$$D(C2_i, C2_j) = norm(C2_i - C2_j) \tag{7.8}$$

Above definitions are for feature extraction over the entire images. We now define distance of two images based on their feature at salient regions. Assume $S_i = (s_{i1}, s_{i2}, \dots, s_{ib})$, $saliency_{s_{ij}} > saliency_{s_{ij+1}}$ be the vector of salient points generated by the saliency detector ordered by their saliency value. Let,

$$W_{ij} = \begin{bmatrix} D(s_{i1}, s_{j1}) & \cdots & D(s_{i1}, s_{jb}) \\ \vdots & \ddots & \vdots \\ D(s_{ib}, s_{j1}) & \cdots & D(s_{ib}, s_{jb}) \end{bmatrix} \tag{7.9}$$

be the matrix of pairs of salient regions with $D(s_{i1}, s_{j1})$ as the distance between the first salient regions of the *i*th and *j*th images. In order to reduce the above distance matrix into a scalar, we need to match the salient regions. One way is to pick the minim distance or the best match between salient regions same as doing to match SIFT features in (7.7) or (7.8) by considering their pixel distances in the images. A more common algorithm is the Hungarian algorithm which takes as input a distance matrix and returns a distance measure as well as the best match. This is a combinatorial optimization algorithm with the benefit of solving the assignment problem in

polynomial time. A better match between two images means lower distance and hence higher similarity.

Repeatability Measure Repeatability indicates how often features repeat in the transformed versions of an image. Inverse of above distances could be used for calculating similarity or repeatability of two images. Repeatability of a set of images is the mean of repeatability of subsequent images (frames) and is defined as (also called tracking repeatability):

$$R_T(I) = \frac{n-1}{\sum_{i=1}^{n-1} D\left(I_i, I_{i+1}\right)} \tag{7.10}$$

where D is one of distance measures defined in section 7.3.1. N is the number of images in a scene class. We compare the repeatability of saliency detector patches at fixed locations over all images and random locations in each image.

7.3.2 Repeatability Results

Results of repeatability and attentive scene classification are shown in this section over a database of natural scenes. This database contains 44 classes each with 24 scenes with large viewpoint changes from locations in a university campus (Jodogne & Piater, 2007). We restrict number of salient points to five (b=5) since the concept of visual saliency is restricted to few features per image by definition. Eight first instances of the seven first classes of the above database with their salient regions marked with rectangles are shown in figure 7.1. As this figure shows, in many cases salient regions are in the same regions in subsequent images but their saliency rank differ. This shows the significance of region matching in attentional scene classification. Features are derived in a window of size (W) around each salient point. Tracking repeatabilities for salient regions for these classes using both features are shown in Table 7.1. As Table 7.1 shows repeatability increases in each class as the number of salient points increase using both SIFT and C2 features.

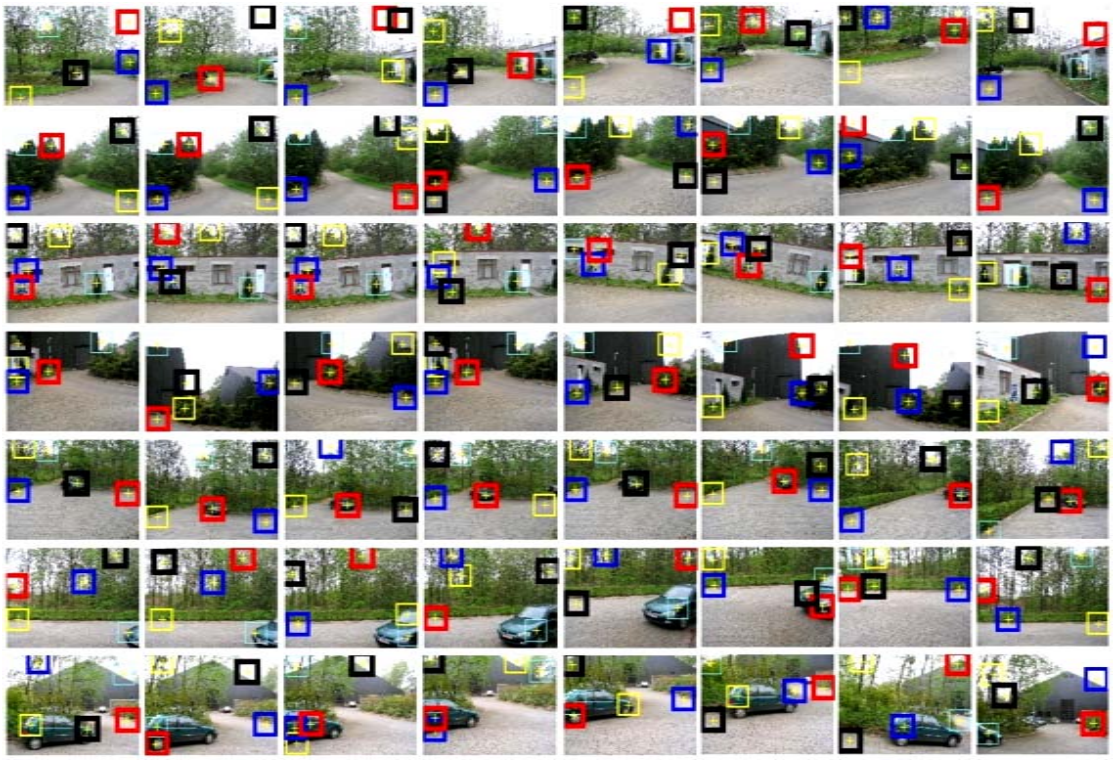


Figure 7.1 Salient regions of 7 natural scenes. Each row is for a different class. Order of salient points is coded by colors: pink, yellow, blue, red and black. Narrowest rectangle is the first and so on. Window size (W) is equal to 60.

Table 7.1 Tracking repeatability of scene classes in Figure 7.1 for different number of salient points. Top and bottom numbers in each row are repeat abilities for SIFT and C2 features, respectively. W is 60 and dimensionality of C2 vector (|C2|) is equal to 4096.

Class No	1	1:2	1:3	1:4	1:5
1	1.35	1.41	1.71	2.27	2.27
	3.24	4.48	5.88	8.05	9.18
2	1.37	1.45	1.62	1.86	1.87
	3.33	4.56	5.64	7.03	8.40
3	1.37	1.44	1.56	1.72	1.77
	3.44	4.70	5.61	6.61	7.50
4	1.36	1.50	1.69	1.89	1.95
	3.23	4.41	5.48	6.43	7.49
5	1.37	1.51	1.67	1.83	1.88
	3.31	4.43	5.53	6.41	7.37
6	1.37	1.50	1.64	1.77	1.82
	3.23	4.38	5.46	6.30	7.14
7	1.36	1.52	1.65	1.76	1.81
	3.29	4.40	5.47	6.22	7.06

7.3.3 Classification Results

Having features and distance measures defined, we are now ready to design the classifier. Here, *k*NN algorithm is used in two cases when features are extracted 1) over the entire scene or 2) only at salient regions. Since using *k*NN is straightforward for the first case, we show how it could be used for classifying scenes in the second case. Attentive scene classification is shown in the pseudo code of Table 7.2.

Table 7.2 Attentive scene classification with kNN

1. Find salient regions for all images in the database.
2. Extract features for all images at salient regions
3. Randomly split images in each class to P for train and remaining Q for test
4. For each test image
    a. For each train image
        i. Calculate the similarity matrix of each salient region of this image to salient regions of a train image using the similarity measures in (x) or (2)
        ii. Calculate the overall similarity between two images by matching the salient regions using Hungarian algorithm
    b. Find the k nearest neighbors to this test image
    c. Assign the class label of this image as the label of the most frequent class
5. Calculate the correct classification rate

In the experiments of this section, we investigate how attention affects learning and recognition of cluttered scenes. Eighteen (p=18) images from each scene class was used for training and the remaining ones (Q = 6) for test. Results of attentive scene classification using SIFT features is shown in Table 3 using distance measure of (7.6) for different number of salient regions and attention window sizes. Classification results using C2 features with dimensionalities 256 and 4096 are shown in Tables 4 and 5 using distance measure in (7.6), respectively.

Table 7.3 Attentive scene Classification using SIFTS with the distance measure in (7.6). Numbers in parentheses are standard deviations over five executions.

W	Number of Salient Points				
	1	1:2	1:3	1:4	1:5
40	15 (1.3)	17.4 (3.2)	16.6 (4.1)	16.2 (1.8)	19 (2.5)
60	29.67 (5.1)	37.75 (3.6)	42 (1.09)	47.6 (2.4)	53.40 (2.4)
100	51.13 (4.2)	74.74 (3.15)	78.25 (3.1)	81.9 (3.5)	85.6 (2)
140	67.67 (1.7)	85.47 (0.5)	90.1 (2.5)	94.5 (1.4)	97.35 (.9)

Table 7.4 Attentive scene Classification using C2 with distance measure in (7.6). Numbers in parentheses are standard deviations over five executions. (|C2| = 256)

W	Number of Salient Points				
	1	1:2	1:3	1:4	1:5
40	13.93 (1.5)	13.18 (.81)	19.01 (1.1)	20.22 (1.9)	21.74 (2.1)
60	25.22 (3.39)	22.95 (2.05)	28.25 (2.5)	31.74 (1.76)	38.48 (2.65)
100	34.46 (2.8)	36.74 (2.6)	48.61 (2.4)	53.40 (2.8)	58.71 (1.7)
140	37.24 (.95)	44.06 (.7)	50.25 (4.4)	58.83 (4.01)	65.27 (2.4)

Table 7.5 Attentive scene Classification using C2 with distance measure in (1). Numbers in parentheses are standard deviations over five executions. (|C2| = 4096)

W	Number of Salient Points				
	1	1:2	1:3	1:4	1:5
40	17.80 (1)	14.26 (2.2)	23.10 (1.65)	25.25 (3)	24.87 (2.1)
60	27.52 (1.3)	26.76 (1.7)	30.93 (1.2)	35.1 (1.9)	43.7 (2.1)
100	40.909 (2.2)	41.7 (3.1)	54.9 (1.8)	60.6 (3)	69.4 (1.1)
140	44.82 (3.2)	53.9 (0.78)	62.75 (1)	70.7 (2.8)	78.28 (2)

As it can be seen from tables 7.3, 7.4 and 7.5, recognition rates increase in all cases when number salient regions increase. This is in accordance with increase in repeatability when raising the number of salient points. We also did recognition using random and fixed regions. Random points were considered to check if high recognition rate is due to overlap in image locations or that is because salient regions repeat in transformed images. Fixed regions were considered to see if salient points are class specific or not. Figure 7.2 compares the classification results using different detectors for window size equal to 60 and different number of salient regions. As its shows saliency works better than random and fixed regions in all conditions over all 44 classes.

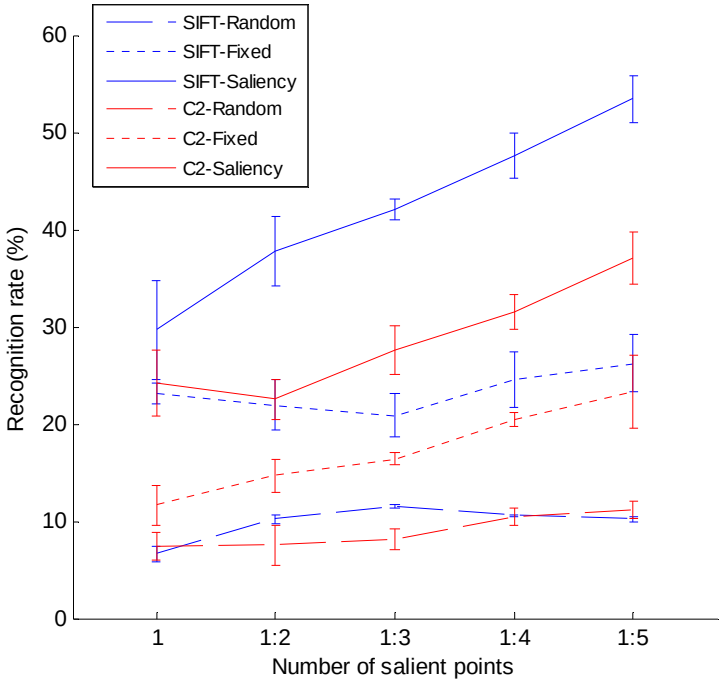


Figure 7.2 Comparison of recognition rate of salient regions as well as other detectors. W = 60 and |C2|=256.

Table 7.6, shows the classification results using KNN with distance measure in (7.7) and SIFTs for different values of parameter σ.

Table 7.6 Classification using distance measure in (7.7) and SIFT features

σ	0	1	2	3	10
Rec. (%)	95.59 (2.1)	97.1 (6.1)	96.77 (2.1)	98.1 (0.37)	93.8 (1.6)

Results are not conclusive for different values of σ. It shows that σ equal to 3 leads to better results in all cases but nothing general can be said on setting this parameter since it shows no trend. Figure 7.3 shows the classification results using KNN with C2 features extracted from the entire image.

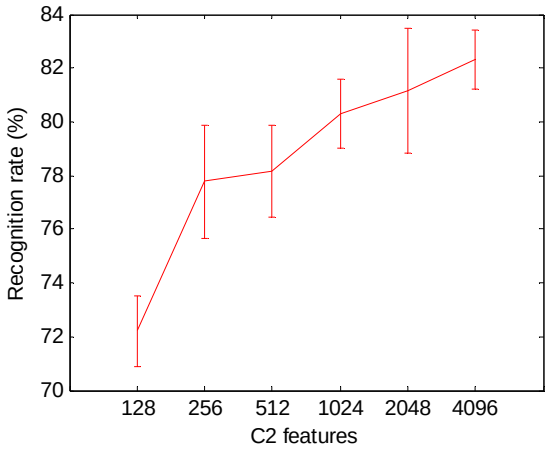


Figure 7.3 Classification results using C2 features with different dimensionalities.

Figure 7.3 shows that increasing the number of C2 features, which means that more detailed and complex structures are extracted from images, significantly enhances the classification result. From the results it can be seen that SIFT features result in higher classification rate compared with C2 features. We also performed classification when feature extraction was done over the entire scene. Classification results using the distance measure in (6) leads to 99.3(.3)%, 82.32(2.1)% and 77.9% recognition rates using SIFT, C2(|C2=4096|) and C2(|C2=256|) features, respectively.

7.4 Attentive and Action-based Scene Classification

Recent trend in robotics is toward developing robots capable of acting autonomously in unknown, stochastic and partially observable environments. This desired quality makes interactive and online learning of visual representations and control policies an essential element. Such dynamic methods result in flexible solutions with less complexity and lower computational power. For a robotic agent to be able to act in a visual environment, it should map its visual perceptual space into physical actions also called visiomotor coordination, vision for action or purposive vision. In contrast with machine vision solutions which assume predefined and rich visual representations in mind of an agent, necessary representations could be learned while the agent performs physical actions for achieving a goal. In this regard vision is formed for the purpose of achieving agent's goals. This is also in alignment of need-based approach theories of visual learning in human brain which states vision is formed to the extent that could handle needs of an agent.

In this regard learning top-down attention is highly coupled with learning representations. Therefore the best way to derive visual attention mechanisms is to learn them in concert with visual representations. Two active but independent research areas have previously been focused on this problem. In first area, purposive vision or vision for action, it has been tried to learn visual representations along with physical actions through automatic state space discretization in reinforcement learning framework. Computational modeling of visual attention and its applications in robotic tasks is the second research area. It serves near the same purpose as above by motivating from attentional mechanisms implemented in humans and animal visual systems.

In this paper, we aim to combine bests of these two worlds to propose efficient algorithms for concurrent learning of representations and actions.

7.4 RLVC

Jodogne et al. (2007) have proposed an approach for learning action-based image classification known as Reinforcement Learning of Visual Classes (RLVC). RLVC consists of two interleaved learning processes: an RL unit which learns image to action mappings and an image classifier which incrementally learns to distinguish visual classes. RLVC could be regarded as a feature based attention method in which the entire image has to be processed to find out whether a specific visual feature exists or not in order to move in a binary decision tree. Like RLVC, our approach also extends the U-Tree to visual domain. Our approach tackles this weakness of the RLVC, the exhaustive search for a local descriptor over the entire image, by computing and searching local descriptors at few spatial locations. Once aliasing is detected a SIFT features is selected which reduces the perceptual aliasing the most. This local visual processing results in very fast scene recognition because features are extracted in small regions and there is no need for exhaustive search for a feature.

The approach that has been followed by RLVC is dividing the visual space when it is necessary. While in ordinary scene classification aim is to achieve maximum classification rate in action-based scene classification aim is to get the highest correct policy rate or reward. Therefore, in this case minimizing the aliasing (equation 2) results in highest correct policy rate (percentage of correct elicited actions by the agent). Aliasing is removed by checking the existence of a learned visual feature each time into two partitions: Those which exhibit that and those which do not. Distinctive visual feature are derived by defining a visual feature and a distance threshold on that. The most widely used visual feature is SIFT features. High dimensionality of SIFT features makes them distinctive, but leads to high complexity in computation which makes it challenging for real world applications like robot navigation.

RLVC generates a binary decision tree in which internal nodes check the existence of a specific SIFT feature in the scene and leaves are states. Employing binary features has the advantages of simplicity and high invariance to image transformations. But require a large number of distinctive features, very high computation to find whether a specific feature exists or not in a scene. For detailed information on RLVC, the reader should refer to Jodogne et al. (2007).

Initially a tree with a single node is created and all the images are mapped to that node. Evidently, such a single state is not enough and aliasing occurs. In each stage of the algorithm, RL is executed for a number of episodes by following a ε-greedy or soft-max action selection policy. Quadruples $(s_t, a_t, r_{t+1}, s_{t+1})$ are only used for Q-table update according to Q-learning update rule:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a)\right) \tag{7.11}$$

While interacting with the environment based on its current policy agent gathers some experiences in each of its leaves. Each memory item under a leaf node could be represented as:

$$B = [b_1, b_2, \dots, b_z], b_i \in \{0,1\}, \quad (7.12)$$
$$F = [f_1, f_2, \dots, f_z]$$

with z as the number of distinct visual features (SIFTs).

Therefore memory is a binary matrix with its columns representing distinctive visual features and rows showing experiences or memory items (captured scene). A "1" in the *ij*th element of the matrix show that the *i*th memory item has the *j*th visual feature (SIFT).

Aliasing in each tree leaf is measured for each memory item and could be derived from the Q-learning formula as:

$$\begin{aligned} Q(s_t, a_t) &= \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right) + Q(s_t, a_t) \\ &= \alpha\Delta_t + Q(s_t, a_t) \end{aligned} \quad (7.13)$$

Each state with $\Delta_t > \tau$ has aliasing and has to be refined (discretized) in order to remove aliasing. Tree refinement or selecting the SIFT feature that mostly reduces the perceptual aliasing is a bit different. The most discriminant feature is calculated as follows:

$$\begin{aligned} f^* &= armin_f \left(var\{y\} - \left(\frac{|L_+|}{|L|} var\{y_a| +\} + \frac{|L_-|}{|L|} var\{y_a| -\}\right)\right) \\ &= armax_f \left(\frac{|L_+|}{|L|} var\{y_a| +\} + \frac{|L_-|}{|L|} var\{y_a| -\}\right) \end{aligned} \quad (7.14)$$

where $\boldsymbol{y_a}$ is the set of all memory items with action a, $y_a| +$ ($\boldsymbol{y_a}| -$) is the set of memory items with (without) feature *f* and action *a*. Sizes of these two sets are $|L_+|$ and $|L_-|$, respectively.

Maximization is done over all pairs of features and actions. A feature which minimizes the variance also has to cluster the images under the node into two populations with significantly different distributions. In our implementation of this case, we used *t*-test for comparing these two distributions. If the power of the *t*-test is below 0.05, then two distributions are considered to be significantly different.

7.4 RLVC with Saliency

Since RLVC checks entire image to find out whether a specific feature exists or not, it could hardly be considered as an attention method (or attentional). In this contribution we propose that instead of all image locations, features could be extracted and searched only in a subset of image locations. These spatial locations should be stable regions. Here we show that regions proposed by the saliency based model of visual attention are nearly stable and result in the same performance as RLVC with the advantage of less computational complexity.

At first, SIFT features of some random images were extracted and then clustered to find distinctive SIFT features. This was done with modified basic sequential clustering algorithm (MBSCA) with different thresholds. Each threshold results in different number of distinctive SIFT features. Threshold is over the Euclidean distance between SIFT features. Here 651 distinctive SIFT feature derived by putting a threshold of 0.8 on the Euclidean distance.

Experiments were done over a 6×6 visual grid (Fig 4) with the goal at position [6 6] and obstacles at [3 3; 3 4; 4 4; 5 2]. In each position agent could observe one image and has not access to it (*x,y*) position. Size of each image is 640×480 pixels.

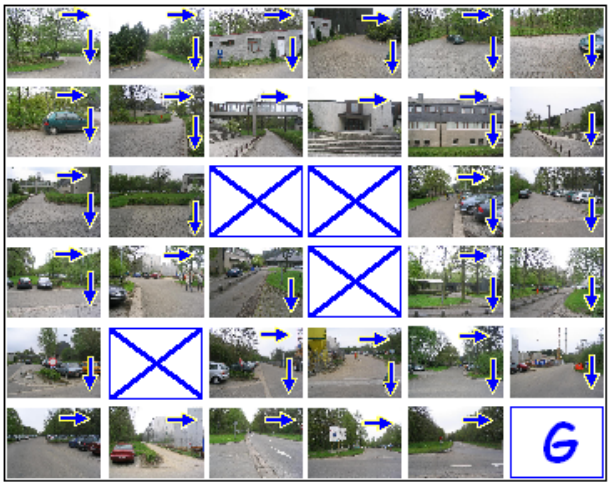
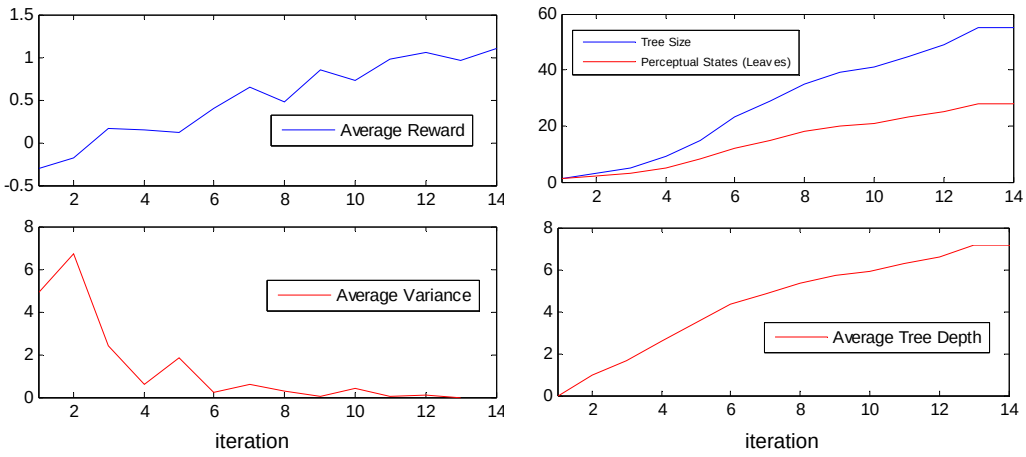
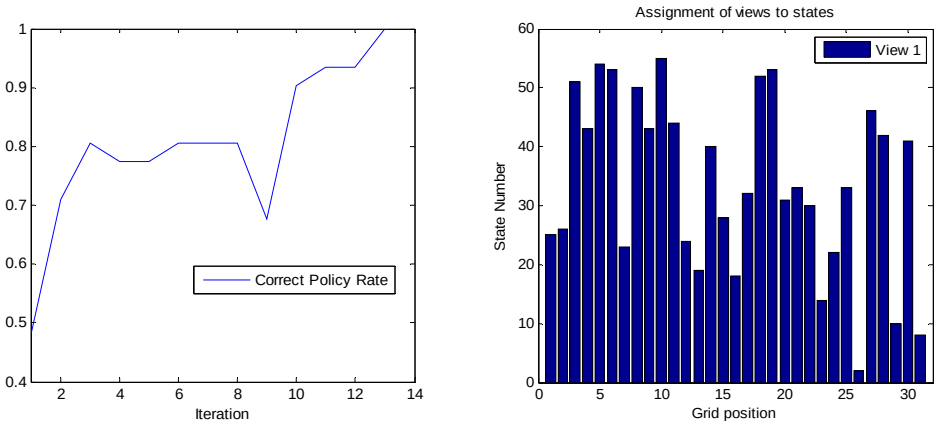


Figure 7.4 Visual Navigation Task 6×6



a) b)



c) d)

Figure 7.5 RLVC Saliency results a) average reward and average variance in Δ_t per iteration b) Tree size, number of perceptual states and average tree depth c) percentage of correct policy rate d) Number of state of each grid cell.

Figure 7.5 shows the results of resolving the visual grid in figure 7.4 when only one view was shown to the agent in each grid position. Agent gets a reward of +10 when it reaches the goal and -1 when it hits the obstacle or escapes from the grid. Any other move bring a 0 reward for the agent. In all experiment alpha = 0.7 and gamma = 0.8, epsilon = 0.7. $\tau = 0.01$. In each grid cell agent could do one of four actions {up, down, left, right}

Increase in reward in Fig. 5.a shows that agent incrementally learns to do the task. And reducing variance of Δ_t means that aliasing is removed until it reaches to 0 which means there is no further aliasing. Shown by Fig.5.b RLVC with saliency generated a binary tree with 54 nodes of which 28 are leaves. This means that 31 distinct scenes are clustered into 28 classes. This clearly shows that for solving action-based recognition it is not necessary to discriminate among all perceptual classes or scenes. For example cells 21 and 25 are classified under the state 33 and because the best action associated for them is the same (down), there is not necessary to discriminate future among these two because there is no aliasing. Fig. 7.5.b. also show that after learning the average tree depth is 7.1 which means that for doing this task it is only necessary to check the existence of 7.1 SIFT features on average only at salient regions.

Correct policy rate in Fig 5.c reaches to 100% when algorithm is finished. A value of 74% at the start is because at first when there is only one node action down is learned by chance which is correct in 74% cases. Assignment of grid cells to leaves of the generated tree is shown in Fig. 5.d.

To compare RLVC and RLVC with saliency we performed the same grid task with the difference that here agent captures of five scenes randomly in each grid cell. These five scenes are from the same class. Both methods successfully solved the task. Assignments of views to states in both approaches are shown in Figure 7.6.

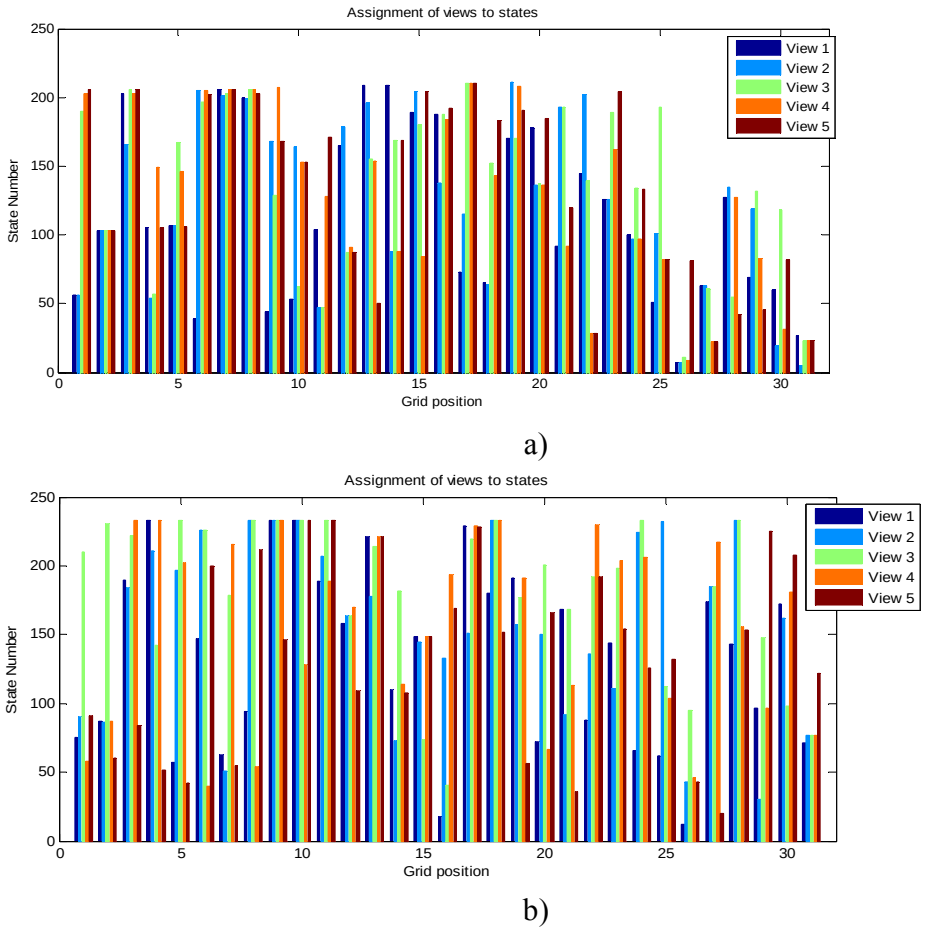


Figure 7.6 Assignment of views to states a) RLVC b) RLVC with saliency.

Detailed comparisons of trees that these two approaches generate are shown in Table 7. To measure the generalization power an index know as *state reduction index* is defined as:

$$SRI = \frac{1}{N}\sum_{n=1}^{N}\frac{\sigma^2\big(V(n)\big)}{mean\big(V(n)\big)},$$
$$V(n) = [v_1(n), v_2(n), \dots, v_5(n)] \qquad (7.15)$$

In above formula, $V(n)$ is the vector of leaf numbers for view of a cell and N is the number of grid cells. Numbering is done at the level order when tree is built. Zero value for *SRI* means that all samples of a scene are classified under the same leaf. *SRI* means high generalization power.

Table 7.7 Comparison of basic RLVC and RLVC with saliency over five view for each grid position and five salient regions.

Window size (w)	**40**	**60**	**100**	**140**	**[640×480]**
Number of states	121.5(6.3)	133(7)	125.5(6.3)	113(1.4)	112(8.4)
Number of nodes	242 (12.7)	265(14.1)	250(12.7)	225(2.8)	223(16)
Avg. tree size	11(0.8)	9.8(0.37)	9.6(0.5)	9.2(0.84)	10.3(0.4)
SRI	0.37(0.1)	0.39(0.1)	0.33(0.08)	0.31(0.004)	0.39(0.2)

As table 7.7 shows as the salient window size grows number of states, nodes, Avg. tree depth as well as SRI reduces which means increasing window size enhances the results. Shown in the last column of Table 7 these values are not much different for both RLVC and RLVC with saliency. This result shows that RLVC with saliency has very much lower computation compared with basic RLVC while having the same performance. This is in accordance with the scene classification result in section 3.

Table 8 compares the effect of increasing the number of salient regions on result. As it shows increasing the number of salient regions only generate trees with lower Avg. tree depth.

Table 7.8 Comparing number of salient regions on trees generated by RLVC with saliency over 2 views for each cell and W = 100.

Salient regions	**1**	**1:2**	**1:3**	**1:4**	**1:5**
Number of states	48 (9.8)	49(5.6)	50.5(2.1)	55(4.2)	50.3(3.5)
Number of nodes	95(19.7)	97(11.3)	100(4.2)	109(8.4)	99.6(7)
Avg. tree size	10.3(2.6)	7.1(0.5)	7.4(0.35)	7.5(0.98)	6.9(0.6)
SRI	0.23(0.004)	0.29(0.01)	0.23(0.04)	0.27(0.02)	0.27(0.05)

7.5 Discussions

In this section, we analyze to what degree attention reduces computations needed for classification. We investigate the computational complexity from two aspects 1) complexity of feature extraction and 2) complexity of feature matching

5.1 Complexity of feature extraction

Assume an image of size $m \times n$, SIFT window of size W and b salient points leads to computational complexity of $C = \frac{b \times W^2}{m \times n}$ feature extraction. C is the ratio of feature extraction at the salient points to the overall image. Since W is usually small compared to m and n, this leads to huge computational saving. Figure 7.7, shows computational saving of feature extraction versus recognition rates for different values of W. Vertical axis shows the ratios of recognition and computational complexity in attentional and without-attention cases. This diagram is plotted for SIFT and C2 features for only first salient region as well as all of them for different window sizes. For example it shows that with a window size of 140 and 5 salient regions, 95% of the recognition in without-attention case (processing entire scene) can be achieved while only extracting features at 31% of the image. Increasing attention window size results in higher

recognition rate and higher computational complexity. An optimal value for window size depends on application and on available computational resources and minimum recognition rate needed. This analysis can also help to determine minimum computational necessary to achieve a maximum recognition rate.

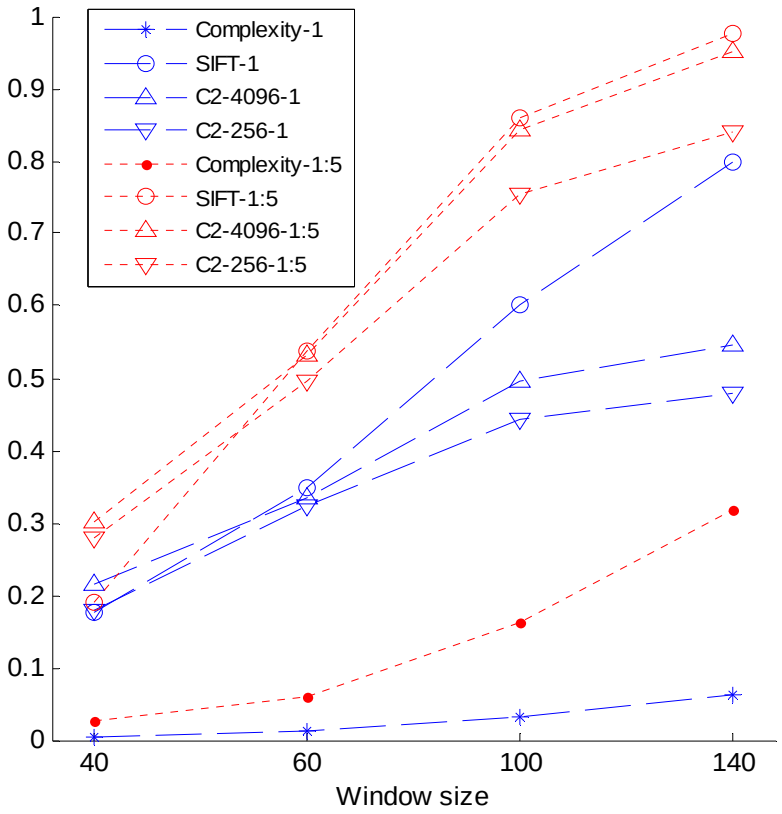


Figure 7.7 Computational complexity vs. recognition rate

5.2 Complexity of feature matching

Restricting feature extraction to a subset of image regions not only reduces the complexity of feature extraction algorithm but it also reduces the complexity of feature matching between two images as we show in this section. Let $\alpha = \frac{W^2}{m \times n}$, be the fraction of image area at a single attention window to the whole image and P be the average number of features per image. As equation (6) shows with P features the complexity of feature matching between two images is P^2. Assuming uniform distribution of features in images, number of features in a single patch is αP. Therefore, according to equation (9) calculating the distance measures among b salient regions of two images (all combinations) and then deriving an overall distance (C'), needs $b^2(\alpha P)^2 + C'$ computations. Since C' is usually small, (matching a few scalars), we could ignore it, thus

$$k = \frac{b^2(\alpha P)^2 + C'}{P^2} \simeq \frac{b^2(\alpha P)^2}{P^2} \simeq b^2\alpha^2 \qquad (16)$$

is the ratio of complexity when features are matched at salient regions versus when they are matched over entire images. Since usually a small value is chosen for b (by definition saliency is

limited to a few salient regions), and α is much smaller than one, then $b^2\alpha^2$ is smaller than one. This means that proposed scene classification approach not only reduces complexity of feature extraction but also it reduces complexity of feature matching.

7.6 Conclusions

This paper has studied the problem of defining and estimating descriptive and compact visual models of scene classes for efficient scene recognition. In accordance with human recognition behavior first a fast, parallel and pre-attentive mechanism processes the entire image and selects some salient regions and then a complex and slow mechanism is restricted on these areas to extract more details and do matching with a database of learned representatives. In the proposed approach, to overcome the bottleneck feature extraction and matching is performed at few spatial locations proposed by visual attention which acts as a front-end to accelerate speed and reduce complexity.

Results show that salient regions have higher repeatability compared with fixed and random patches in instances of a scene class. This means that it is not the overlap among regions which is responsible for similarity between two images, rather it is because the positions are at near the same areas but with their ranks misplaced. As window size increase similarity and hence recognition rate over both feature types. SIFT features seem to more successful in average. Compared with the situation when features are extracted over all images, we were able to achieve near and sometimes higher recognition rate with smaller amount of computation.

We aimed to show that high recognition is still possible without processing the entire image at salient points. Saliency model is purely data-driven and simply selects some spatial locations without using any feedback mechanism or top-down gains. Modifying the attention systems to generate robust salient points when viewpoints change is a future research area. For example adding top-down capabilities to the model biases it to select the same spatial areas in images and therefore leads to higher stability in images. This also copes with the saliency for recognition hypothesis claimed by Gao et al. (Gao et al., 2007). Applying this approach to robot control because viewpoints there change gradually also might lead to great saving in computation]. Repeatability of salient points under different image transformations should be analyzed.

Chapter 8
Interactive Learning of Space-based Visual Attention Control and Physical Actions

Ali Borji, Majid N. Ahmadabadi, Babak N. Araabi

Published in:

IEEE International Conference on Robotics and Automation ICRA 2009

Abstract. This paper introduces a new method for learning top-down and task-driven visual attention control along with physical actions in interactive environments. Our method is based on the RLVC algorithm and adapts it for learning spatial visual selection in order to reduce computational complexity. Proposed algorithm also addresses aliasings due to not knowing previous actions and perceptions. Continuing learning shows our method is robust to perturbations in perceptual information. Proposed method also allows object recognition when class labels are used instead of physical actions. We have tried to gain maximum generalization while performing local processing. Experiments over visual navigation and object recognition tasks show that our method is more efficient in terms of computational complexity and is biologically more plausible.

Keywords: State space discretization, Top-down attention control, Visual attention, Active perception, Spatial configuration, Object recognition, Scene understanding, Reinforcement learning

8.1 Introduction

Similar to humans and animals, artificial creatures like robots are limited in terms of allocating their resources to huge perceptual information. That is mainly because of the serial processing mechanisms used in the design of such creatures which allows processing of only a small amount of incoming sensory information at any given time. Visual attention mechanisms serve to implement a bottleneck through which only informative and relevant information are allowed to pass to higher level visual processing centers. Since robotic agents are usually supposed to

guarantee a limited response time, attention is an efficient solution in this area as in biological systems.

To perform a task, agents should be able to perceive the environment and perform appropriate physical actions. Perceptual actions are available in several forms like where and what to look in visual modality. The main concern in learning attention is how to select the relevant information, since relevancy depends on the tasks and goals. In this study, we consider task relevancy of visual attention and aim to extract spatial locations which help the agent to achieve its goals faster.

It is important that a solution for learning task-based visual attention control to take into account other relevant and interleaved cognitive processes like decision making, action selection, memory, etc. There are several biological evidences for this. It has been previously shown that attention and eye movements are context-dependent and task-specific (Yarbus, 1967). Previous experiences also influence attentional behaviors indicating that attention control mechanisms can be learned (Maljkovic & Nakayama, 1994). Some neuropsychological evidences suggest that human beings learn to extract useful information from visual scenes in an interactive fashion without the aid of any external supervisor (Gibson & Spelke, 1983; Tarr & Cheng, 2003). Instead of attempting to segment, identify, represent and maintain detailed memory of all objects in a scene, there are evidences that claim our brain may adopt a need-based approach (Triesch et al., 2003), where only desired objects are quickly detected, identified and represented. Considering above evidences, we introduce a model to consider the influences of task, action, learning and decision making to control top-down visual attention.

From another perspective learning top-down attention is highly coupled with learning representations. Therefore the best way to derive visual attention mechanisms is to learn them in concert with visual representations. Two active but independent research areas have previously been focused on this problem. In first area, purposive vision or vision for action, it has been tried to learn visual representations along with physical actions through automatic state space discretization in reinforcement learning framework. Computational modeling of visual attention and its applications in robotic tasks is the second research area. It serves near the same purpose as above by motivating from attentional mechanisms implemented in humans and animal visual systems. In this paper, we aim to combine bests of these two worlds to propose efficient algorithms for concurrent learning of representations and actions.

8.1.1 Purposive vision or Vision-for-Action

Humans are remarkably efficient to act reasonably in an environment based on perceptual information they receive. Vision is the most important sensor humans rely on, and that's the main reason of being the most-studied mode of machine perception in artificial intelligence (AI). Despite the huge active research in computer vision and robotics, many real-world visumotor tasks which are easily performed by humans are still unsolved. Of special interest is designing efficient learning algorithms, in terms of high accuracy and low computational complexity, for enabling autonomous mobile robots to act in visual interactive environments. This is actually much harder to tackle compared with controlled visual environments (e.g. indoor scenes) used in robotic laboratories. Example applications of visual learning are vision guided navigation, vision-based place recognition, grasping and object manipulation.

Recent trend in robotics is toward developing robots capable of acting autonomously in unknown, stochastic and partially observable environments. This desired quality makes

interactive and online learning of visual representations and control policies an essential element. Such dynamic methods results in flexible solutions with less complexity and lower computational power. For a robotic agent to be able to act in a visual environment, it should map its visual perceptual space into physical actions also called visiomotor coordination, vision for action or purposive vision. In contrast with machine vision solutions which assume predefined and rich visual representations in mind of an agent, necessary representations could be learned while the agent performs physical actions for achieving a goal. In this regard vision is formed for the purpose of achieving agent's goals.

A schematic view of our framework for learning vision for action is shown in Fig. 8.1. A scene is captured by the visual sensor (eg. CCD camera) of the agent (early vision) and then undergoes a number of higher processing and usually computationally intensive processes like recognition, segmentation, etc. From this information, state of the agent is derived (higher vision). The decision making and attention control module concurrently controls top-down attention and motor actions. The attention control signal generated by this module controls physical properties of the visual sensor, e.g. shifts the eye to new positions (overt attention) or tune high-level visual operations. The motor actions are executed by the body of the agent and affect the world. A critic who is aware of the dynamics of the world evaluates the effect of this action on the world and generates a scalar reward which is used by the learning system for tuning the decision making unit. Actually, this unit performs credit assignment (reward distribution) among top-down attention control and decision making modules. Decision making and attention control module, change adaptively such that the total received reward is maximized.

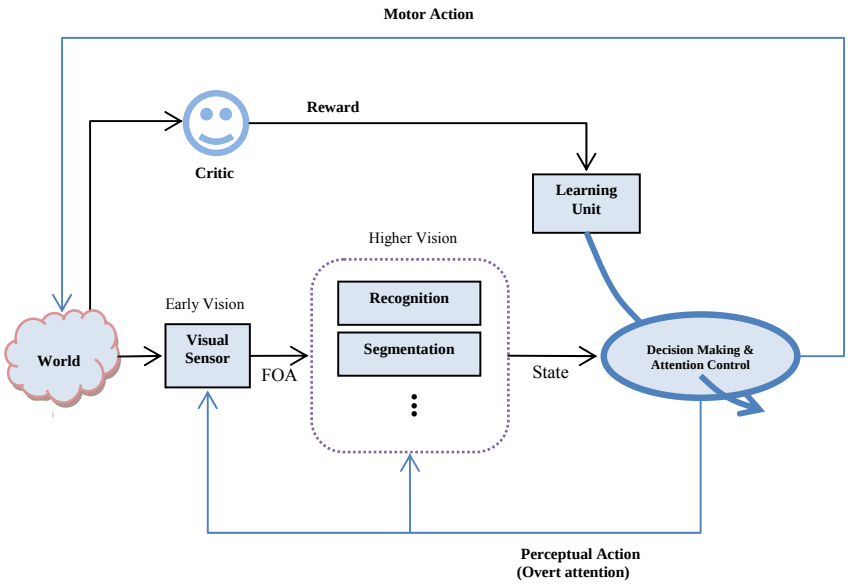


Figure 8.1 Schematic view of the learning system for purposive vision through visual attention. FOA stands for focus of attention

8.1.2 Visual Attention

Visual attention is the capability of biological visual systems to select a subset of task-relevant visual information and gate it to higher level more expensive processing centers. It acts like a filter in which some spatial regions are fast and efficiently selected and other expensive operations are then focused on these areas. Extending this idea, attention can act as a hierarchy where in upper layers of this hierarchy, operations have lower complexity and as we move toward the end, operations become more expensive. Selection of a subset of image regions could be space-based, object based or feature based. This explanation is illustrated in Fig. 8.2.

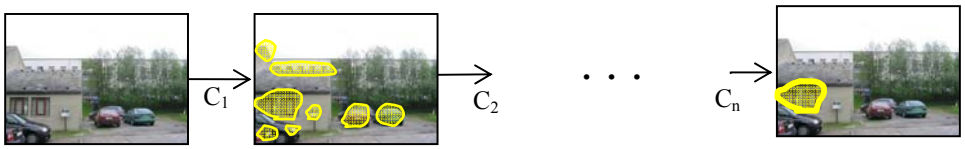


Figure 8.2 Illustration of visual attention process. Selection criteria gradually become more complex.

For a system to be attentional, it is necessary that the cost (complexity) of selection process plus the operations over the final selected regions (C_a) to be less than the operations over the entire image (C_b).

$$C_a = \sum_{i=1}^{n} C_i + C_e \tag{8.1}$$

where C_i is the cost of selection process in the *i*th stage and C_e is the cost of the operations over final selected regions by the attentional system. Therefore, $C_a < C_b$ is the condition of attention provided that both systems could solve the task equally (i.e. they have near the same performance over other indices measured by task performance). Selection of image regions could be done in a single shot in parallel or sequentially like what happens in human saccade behavior.

8.1.3 Contributions

Motivated by human saccadic behavior, several contributions are proposed in this paper for learning purposive vision strategies. The key idea is that state space is incrementally grown or shrank whenever perceptual aliasing occurs by looking at distinctive spatial locations. Details of our contributions are reviewed in the following.

Saccade Tree (S-Tree) S-Tree like RLVC (Jodogne et al. (2007)) is built on top of U-Tree algorithm. RLVC could be regarded as a feature based attention method in which the entire image has to be processed to find out whether a specific visual feature exists or not in order to move in a binary decision tree. Our approach tackles this weakness of the RLVC, the exhaustive search for a local descriptor over the entire image, by computing and searching local descriptors at few spatial locations. Once aliasing is detected a spatial location is selected which reduces the perceptual aliasing the most. This local visual processing results in faster scene recognition

because features are extracted in small regions and there is no need for exhaustive search for a feature.

Handling POMDP cases In real world applications there are situations when perceptual information is the same but different actions has to be done. Actually by looking at other sensors or other visual areas aliasing could not be overcome because sensors get the same exact perceptions in several but different states. In such situations agent has to refer to its previous actions or observations. We show that saccade tree algorithm is capable of resolving POMDP cases by checking previous actions or observations in internal tree nodes.

Object Recognition and GIST Proposed saccade tree method could be used for object and scene recognition. Here the reward is only correct prediction of the object or scene class. Definition of aliasing also differs from the aliasing when actions have to be done. GIST is the capability of human beings to judge about the category of a visual scene very fast (about 80ms). It means that before starting to saccade, a rough estimation of an object or scene class by looking at the whole image (but very fast and low complexity) is performed to bias subsequent saccades. The same is done in Saccade tree where first a very low complexity operation is done in the root of the tree and then in order to classify a scene some spatial locations are investigated. This results in faster state determination.

Invariance Analysis The main concern in the proposed approach is its capability to generate less number of states or state abstraction. This is the same statement as generalization in context of computer vision. In this section we analyze the invariance properties of saccade tree algorithm to image transformations and propose solutions to enhance its generalization. Furthermore, robustness of the saccade tree is also analyzed with respect to perturbations in perceptions.

8.2 Related Works

Related works to our approach are reviewed from two perspectives. First subsection reviews works based on reinforcement learning dealt with state space abstraction or discretization. Second category are approaches for modeling visual attention mainly those concerned with learning visual attention control.

8.2.1 State- Space Discretization

Several approaches for interactive discretization of state space have been proposed. Techniques using a non-uniform discretization are referred to as variable resolution techniques (Munos & Moore, 2002). The parti-game algorithm (Moore & Atkeson, 1995) is an algorithm for automatically generating a variable resolution discretization of a continuous, deterministic domain based on observed data. This algorithm uses a greedy local controller and moves within a state or between adjacent states in the discretization. When the greedy controller fails, the resolution of the discretization is increased in that state. The G algorithm (Chapman & Kaelbling, 1991), and McCallum's U-Tree algorithm (1996), are similar algorithms that automatically generate a variable resolution discretization by re-discretizing propositional techniques. Like parti-game, they both start with the world as a single state and recursively split it when necessary.

The continuous U-Tree algorithm described in (Uther & Velso, 1998), extends these algorithms to work with continuous state spaces.

U-Tree The U-Tree abstracts the state space incrementally. Each leaf l_t of the U-Tree corresponds to an abstract state s_t. Leaves store the action-values $Q(s_t, a_t)$ for all available actions a_t. The tree is initialized with a single leaf, and new abstract states are added if necessary. Sub-trees of the tree represent subtasks of the whole task. Each sub-tree can have other sub-sub-trees that correspond to its sub-sub-tasks. The hierarchy breaks down to the leaves that specify the primitive sub-tasks. The procedure for construction of the abstraction tree loops through a two phase process: *sampling* and *processing*. During the sampling phase the algorithm behaves as a standard RL, with the added step of using the tree to translate sensory input to an abstract state. A history of the transition steps, i.e. $T_t = (s_t, a_t, r_t, s_{t+1})$ composed of the current state, the selected action, the received immediate reward, and the next state is recorded. The sample is assigned to a unique leaf based on the value of the current state. Each leaf has a list per action for recording the sampled data-points. After some learning episodes the processing phase starts. In this phase a value is assigned to each sample:

$$\begin{aligned} V(T_t) &= r_t + \gamma V(\bar{s}_{t+1}), V(\bar{s}_{t+1}) \\ &= \max_a Q(\bar{s}_{t+1}, a) \end{aligned} \tag{8.2}$$

where $\bar{s}_{t+1}$ is the abstract state that s_{t+1} belongs to. If a significant difference among the distribution of sample-values within a leaf is found, the leaf is broken up to two leaves. To find the best split point, the algorithm loops over the features. The samples within a leaf are sorted according to a feature, and a trial split is virtually added between consecutive pairs of the feature values. This split divides the abstract state into two sub-sets. A splitting criterion compares the two sub-sets, and returns a number indicating the difference between their distributions. If the largest difference among all features is bigger than a confidence threshold, then the split is introduced.

RLVC RLVC shares the same theoretical basis as the original U-Tree algorithm. It iteratively builds a series of classifiers $C = \{C_1, C_2, \dots, C_k\}$ that discretizes the perceptual space P into a finite set of perceptual classes by testing the presence of features in the percepts. After k refinement steps, the classifier denoted by C_k, partitions the visual perceptual space P into a finite number m_k of *visual classes* $\{v_1^k, \dots, v_{m_k}^k\}$. C_k is a binary decision tree, each internal node of which tests the presence of a given visual feature, and each leaf of which corresponds to a visual class. In iteration k, a visual class, v_i^k, is refined if it has perceptual aliasing meaning that optimal decisions cannot be always made since different percepts requiring different actions are grouped together. In order to reduce aliasing, visual classes with aliasing are expanded by checking the existence of a feature. To detect nodes with aliasing *Bellman's Residual* or *TD-error* is measured for all experiences under a visual class which is defined as:

$$B_k(s,a) = \mathcal{R}(s,a) + \gamma \max_{a' \epsilon A} Q_k'^*\big(C_k\big(\mathcal{T}(s,a)\big), a'\big) - Q_k'^*(C_k(s), a) \tag{8.3}$$

where $Q_k'^*$ and B_k are the estimation of the optimal Q function using C_k and the residual at state s, respectively. A non-zero $B_k(s,a)$ for some time stamp t indicates the presence of aliasing in the visual class C_k (s) with respect to the action a. After aliasing is detected a SIFT feature (Lowe, 2004) from the percepts in C_k (s) is selected which mostly reduces the variance of Bellman residuals and also results in two significantly different distributions.

8.2.2 Attentional Studies

A large number of computational models of visual attention or their applications in computer vision or robotics have already been proposed. Some studies have tried to model bottom-up task-independent component of visual attention by proposing hypotheses by them human vision might select visual information like saliency or entropy, etc. Other methods have tried to model top-down task specific component of visual attention. Here we also try to derive a top-down closed loop attention model in an interactive fashion. Some models have been based on neural models which are not the focus of this paper. Since our approach is based on reinforcement learning we first review attentional models based on RL and then some other relevant studies.

Approaches based on RL Reinforcement learning has previously been used for learning visual attention control in robotics and machine vision especially for applications like scene classification and robot localization. In (Fritz et al., 2004, Paletta et al., 2005), a 3-step architecture is proposed for an object recognition task. First, it extracts potential focuses of interest (FOI) according to an information theoretic saliency measure. Then it generates some weak object hypotheses by matching the information at the FOI's with codebooks. The final step is done using Q-learning with the goal of finding the best perceptual action according to the search task. In (Gonic et al., 1999.a), two approaches are proposed in a robotic platform with neck, eyes and arms for attention control. The first approach is a simple feedforward method which uses back-propagation learning algorithm while the second one uses reinforcement learning and a finite state machine for state space representation. The robot has 3 types of actions: attention shift, visual improvement and haptic improvement. Their results confirm that the second approach generates a better performance in terms of finding previously observed objects even with fewer movements in head and neck and also in attention center shift. In (Gonic et al., 1999.b), another robotic platform containing articulated stereo-head with 4 degrees of freedom is presented which can select the region of interest, perform attention shift with saccadic movements, build a map out of the environment and update it according to current observation. In (Gonic et al., 1999.b), attention control has two steps: first, coarse eye movements and then more precise iterative adjustments around the first points. The termination condition of this process is to reach a maximum correlation among what it finds and what it expects.

An RL approach for learning gaze control for a mobile robot performing a visual search task is proposed in (Minut & Mahadaven, 2001). This model is implemented using a fixed pan-tilt-zoom camera in a visually cluttered lab environment which samples the environment at discrete time steps. The agent has to decide where to fixate next, merely based on visual information, in order to reach the region where a target object is most likely to be found. The model consists of two interacting modules. In the first module, RL learns a policy on a set of regions in the room for reaching the target object, using an objective function which is the expected value of sum of discounted rewards. By selecting an appropriate gaze direction at each step, this module provides

top-down control in the selection of the next fixation point. The second module performs "within fixation" processing, based exclusively on visual information.

In (Reichle & Patryk, 2006), RL is used for modeling the behavior of an expert reader by predicting where eyes should look and how long they should stay there for achieving best comprehension from the text. This work is interesting as it models attention as an optimization problem and then uses RL to solve it.

Other studies An important evidence from biology reported in (Maljkovic & Nakayama, 1994), states that attention could be learned by past experience. In a behavioral task, human subjects were supposed to answer a question about a quality of a specific visual item in a synthetic visual search scene. Subjects had lower reaction times when quality of the object stayed the same during successive trials. This study shows that subjects developed a memory during the task. A modeling work trying to explain such behavioral data is done in (Mozer et al., 2006). They have proposed an optimization framework to minimize an objective function which is a sum over reaction time in each state weighted by the probability of that state to occur. Then using a Bayesian Belief Network (BBN), they solved that minimization problem. These results encourage using a learning approach for attention control in AI.

Many models of visual attention in AI are based on the saliency-based model (Itti et al., 1998) which is an extension and implementation of an earlier model of visual attention introduced by Koch and Ullman (Koch & Ullman, 1985). This model is based on the saliency concept and mimics the overall structure of the early visual system for detecting the locations which convey more visual signals and are different from their surroundings. For example a red dot is more discriminant than a blue dot in a white background containing blue dots. Simplicity and little computation are the two main advantages of this model. It has been continuously updated and is the basis of the newer models. It has also been used to explain behavioral data on simple synthetic and static search arrays to dynamic natural stimuli like movies and games (Peters & Itti, 2008). Navalpakkam and Itti (2005) have proposed a model for how a specific task at hand can influence attention, and try to address how the top-down signals can be learned in a supervised manner. They hypothesize that a top-down task-relevance map may be encoded in the brain additional to the visual saliency map, and the final eye movements may be generated by integrating information across theses two saliency maps to form an attention guidance map. In their model, a vector of feature weights (one weight per feature) is learned from images containing the target. This way, task-driven saliency is computed and combined with visual saliency to deploy attentional shifts. Unlike Navalpakkam and Itti (2005), our approach considers top-down influences of visual attention interactively which could be applied to a wider range of tasks rather than simple string matching and supervised learning. This approach is more general because it allows the agent to interact with the environment and to find its own way of achieving a goal.

In (Walther & Koch, 2006), Walther at al. have proposed an approach for learning and recognition of multiple objects in cluttered scenes using the saliency-based model of visual attention. They have modified the saliency model to find spatial salient regions which are more likely to contain an object. That way, the modified model could consider the extent of the objects at the focus of attention. Then they have used the SIFT features (Lowe, 2004) for recognition of multiple objects in the scene. By comparing the performance of David Lowe's recognition algorithm, with and without attention, they have shown that their approach can enable one shot

learning of multiple objects from complex scenes, and that it can strongly improve learning and recognition performance in the presence of large amounts of clutter.

In (Siagian & Itti, 2008), a context-based outdoor scene recognition algorithm for mobile robotic applications is proposed based on the idea of "gist" on top of the saliency model. This method is claimed to have low computational complexity while being biologically plausible.

There are also other computational models of visual attention. Deco and Schurmann (Deco & Schürmann, 2000) modulated the spatial resolution of the image based on a top-down attentional control signal. Tsotsos et al. (Tsotsos et al., 1995) used local winner-take-all networks (WTA) and top-down mechanisms to selectively tune model neurons at the attended location. Making extensive use of feedback and long-range cortical connections, Hamker (Hamker, 2005.a, Hamker, 2005.b) modeled the interactions of several brain areas involved in processing visual attention, which enables them to explain some physiological and behavioral data. Sun and Fisher (Sun & Fisher, 2003) have developed a framework for object-based attention using "groupings". Presented with a manually preprocessed input image, their model replicates human viewing behavior for artificial and natural scenes.

8.3 Learning top-down space-based visual attention control

Our proposed model for learning sequential visual attention control is explained in this section. Agent operates in an environment and has access to it only through its embedded camera. It affects the world by performing a set of physical actions. Agent's main aim is to learn vision to action mapping though interacting with the environment. To this end, agent builds a tree structure which allows forming visual representations and deriving attention. Visual attention implements a fast mechanism for the agent to find its states. In multimodal case, attention interprets as the order of turning the sensors on sequentially. If agent has a single sensor, like camera, attention is interpreted as selecting a subset of information captured by that sensor.

8.3.1 Basic Method

An example scenario of the proposed model is as follows. Agent captures a visual stimulus. An early vision unit, which is equivalent to early visual cortical areas V1 and V2, processes the image in order to detect its salient locations or extracting visual features. This unit only processes the image at a small subset of spatial locations instead of a holistic processing. Output of this unit is the visual content at the focus of attention (FOA). Information at the FOA is then transmitted to a higher cognitive area which corresponds to higher visual system areas like V4, IT, PFC, etc. Processing at this stage could be identifying an object, returning the cluster of a visual feature, etc. This is in accordance with space-based theories of visual attention (Posner & Cohen, 1984). These two stages perform local processing and ignore global content of the image. Another top-down component is the "GIST extractor " which in psychophysical terms is the ability of people to roughly describe the type and overall layout of an image after only a very brief presentation, and to use this information to guide subsequent target searches (Oliva & Torallba, 2006). Higher cognitive processes which are believed to occur in higher visual areas V4/IT perform a variety of cognitive visual processes like scene understanding, object recognition, etc. Figure 8.3 shows the overall system structure for learning top-down attention.

With each eye movement, agent moves one step down in the attention tree. Agent successively attends to spatial locations until it reaches to a leaf in the tree. (The loop among

saliency detector, codebook database and attention tree in the Fig 8.3). GIST extractor unit performs a fast global operation over the image and acts like a heuristic to reduce the number of internal tree nodes that should be examined by focusing the search over a subtree. It's important to note that information extracted from the GIST extractor unit is not rich enough to determine the exact category of the image and it only gives rough cue toward that. This is equivalent to overt attention and eye movements in context of behavioral studies.

Agent performs a physical action according to its current learned policy. Agent should learn building attention tree (forming visual representations) and learning percepts to actions in parallel. A simplification is to decouple these parameters and follow a quasi-static approach to solve the whole problem. Each time, for a period of iterations, one component is hold fixed and the other one is updated until attention tree is fixed and RL policy is learned.

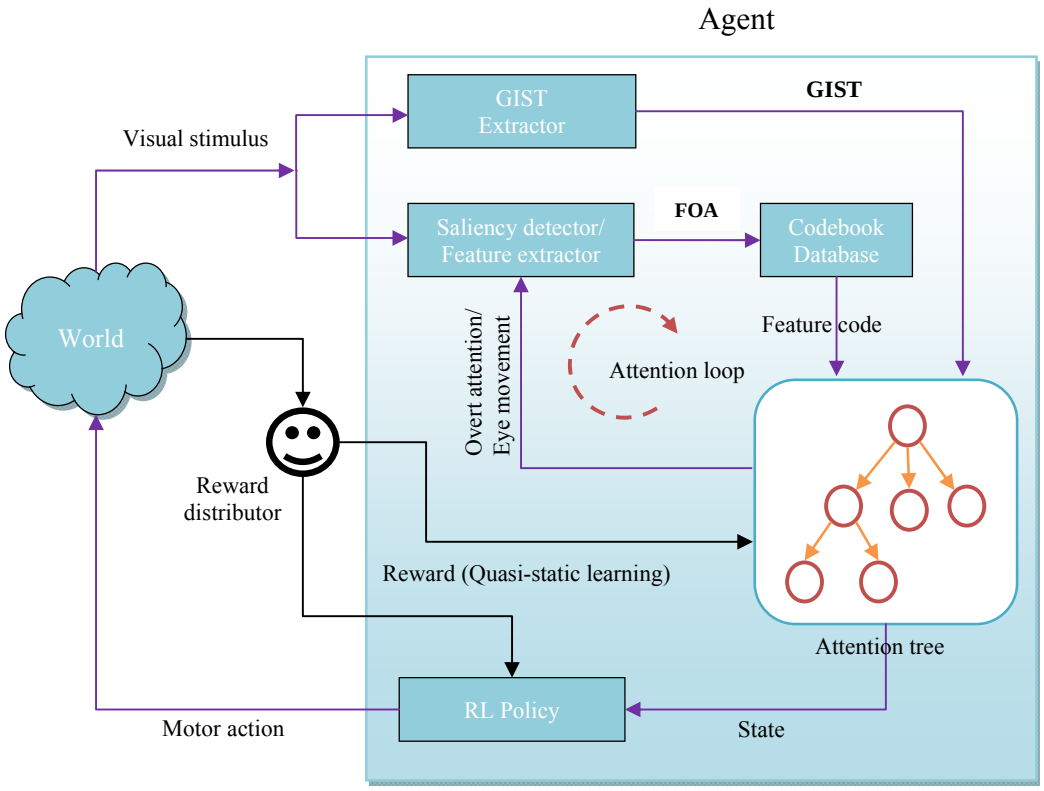


Figure 8.3 Overall architecture for learning concurrent spatial visual attention and physical actions

8.3.2 Visual Representations

Only few portions of an image contain relevant information for the task to be done. An efficient approach in interactive vision systems would be to extract visual features or spatial regions that convey the informative content of the raw images. Visual features lead to a mid-level representation of the images that is halfway between the raw images and a more semantic representation of the observed objects. A visual feature is a vector of real numbers that represents some essential, discriminant and meaningful portion of an image. Visual features have a much smaller dimensionality than the image space, targeting a compact representation that improves

and speeds up the image analysis. Furthermore, though it is not strictly required, visual features often add some invariance to viewpoint or illumination changes, with respect to the raw pixels of the original image.

Methods for representation of images can be divided into two broad categories: Global appearance and local appearance methods. Global appearance methods consider the entire set of image pixels. Some global features are normalized images, Eigen-Patches and Histograms (Histogram-based approaches). On the other hand, local appearance methods identify statistically or structurally significant portions of the images then represent each selected image portion as a vector of real numbers that is known as a local descriptor. The problem of matching images is consequently reduced to that of matching the local descriptors. Advantages of local-appearance methods over global appearance methods are 1) There is no need for 2D or 3D models of objects 2) Local-appearance is robust to partial occlusions and 3) No segmentation is required. Local-appearance methods have been used in a broad class of computer vision problems, such as image classification and categorization, object detection and recognition, camera calibration, stereovision, structure from motion, robot localization and tracking (Mikolajczyk & Schmid, 2005).

In practice, local-appearance methods involve two elements: The interest point detector selects the most informative and reliable portions of an image and then local descriptor generator converts the selected portions to a vector of real numbers that characterize the appearance of these portions. The portions selected by the interest point detector are centered on a so-called interest point in the image, and the portion itself is called the neighborhood of the interest point. The most widespread descriptors (Mikolajczyk & Schmid, 2005) are Filter banks (Like gabors and Wavelets, textures, etc), SIFTs, Gradient Moment Invariants, Spin Image, Complex Filters and Cross Correlation. Visual tasks like image classification using these features can be achieved through at least two distinct approaches: by matching the visual features, or by learning a predictive model. Here we follow local appearance based a feature matching approaches.

A more efficient method which leads to fewer if conditions (i.e tree depth) is to build trees in which more than two branches exist in internal tree nodes. Compared with binary trees employed by RLVC, these trees lead to lower computation by looking at specific spatial locations, match with biology and saccadic eye movements but have lower generalization.

Sequential attention in our method shifts the focus of attention toward the discriminant spatial locations. Final output of the algorithm is a scanpath of eye movements. There is no need that the pattern at each FOA to be represented in fine detail, but an approximate characterization also suffices to discriminate among objects or scenes. Let $f: P \rightarrow T$ be a mapping from the set of focuses of attention P to a discrete set of features T. Here, f returns the class of a specific SIFT feature or a set of features at a member of P which is a circular or rectangular region. In order to derive a rough local descriptor representation, SIFT features of some random images $Q = \{q_1, q_2, \dots, q_{|Q|}\}$ are extracted and then clustered using standard k-means clustering algorithm. Therefore a set of $|T|$ clusters $T = \{\tau_1, \tau_2, \dots, \tau_{|T|}\}$ hereafter called codebooks are generated. The specific feature that f returns its codebook could be the closest SIFT to the center of FOA, f_1, or the SIFT feature with highest magnitude among SIFT features at FOA, f_2. Another possibility is to cluster the histogram of SIFT features at the attended locations. In this case, some sample histograms could be clustered at the start of the algorithm to create codebooks or incrementally through learning the tree from memories under a node. Since this is a complex operation and our

experiments showed that it does not enhance the results (generalization invariance), we only use the above two functions for visual representation. An example of the visual attention tree is shown in figure 8.4.

$f_1 = argmin_j \parallel SIFT_{FOA,k} - \tau_j \parallel, k$ is the closest SIFT feature to the center of FOA

$$f_2 = argmin_j \parallel SIFT_{FOA,k} - \tau_j \parallel, k = argmax_i \parallel SIFT_{FOA,i} \parallel \quad (8.4)$$

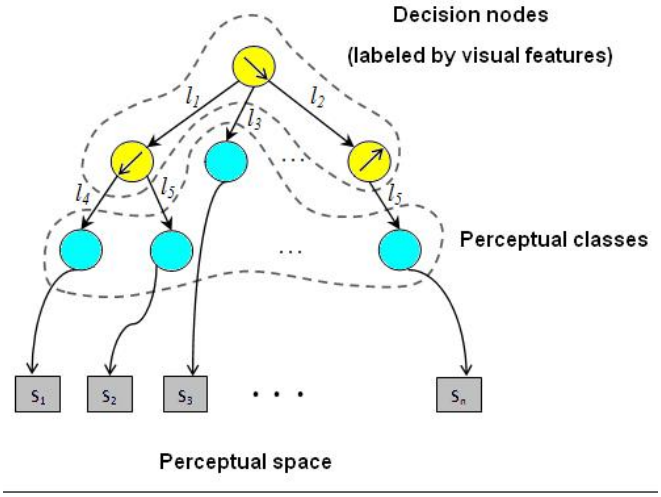


Figure 8.4 Visual attention and state space discretization. with multi-valued features employed in this paper. Internal nodes check features and leaves are perceptual states.

When growing the tree in order to discriminate among perceptual inputs (i.e to reduce perceptual inputs) a next spatial location has to be checked in the tree. Next saccade location can be selected relative to the current location or in random. Saccade region could be either circular or rectangular. While adopting a circular region seems to be more biologically plausible, here we use a rectangular region due to ease of implementation and speed.

Figure 8.5 shows sample objects form COIL 100 object database along with their SIFT features extracted (Link to COIL 100 DB). Eight major saccade directions are shown in fig 8.5.b.

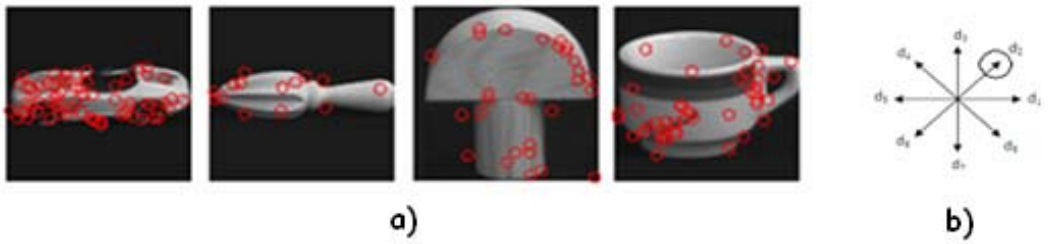


Figure 8.5 *a)* Sample objects from COIL 100 object database with extracted SIFT features. *b)* Discretization of the angle for encoding of saccadic movements ($\Delta\alpha = 45^o$).

8.3.3 Learning top-down attention tree

In our method, attention is directed toward spatial regions. An attention tree (saccade tree or S-Tree) is incrementally built from the incoming visual inputs. In each node of this tree, visual content at the focus of attention (FOA) is inspected. To encode the visual content at the focus of

attention we use the SIFT descriptors which have shown to be very useful for object and scene recognition and image stitching. Before learning S-Tree, SIFT features are grouped into some clusters.

An efficient way to implement attention and state space construction is by means of tree data structures. They are readable by humans and are hierarchy structured which makes them a suitable mean for deriving state-space abstraction and inclusion of the available knowledge for an RL agent.

Visual discretization is performed via saccade tree whenever aliasing occurs. Such refinement is performed to increase the cumulative reward of the agent. Each internal node of the S-Tree proposes a single saccade toward a specific spatial location. Edges below a node test the codebook of a SIFT feature. Based on the observed codebook, next saccade is initiated until a leaf node is reached. Leaves point to states in the Q-table. Pruning is done when algorithm ends for instance by merging the nodes with the same best actions or replacing nodes with all their leaves having the same best actions.

S-Tree is incrementally built in a quasi-static manner in two phases: 1) *RL-fixed (Tree-Update)* phase and 2) *Tree-fixed (RL-Update)* phase. The algorithm starts with one node in the Tree-fixed phase. In each phase of the algorithm external feedback of the critic, in the form of a scalar reward or punishment, is alternatively used to update the Q-table or refine the attention tree. Initially a tree with a single node is created and all the images are mapped to that node. Evidently, such a single state is not enough and aliasing occurs. Then, the algorithm breaks the node into a number of leaves based on some gathered experiences under it. In each Tree-fixed phase, RL algorithm is executed for a number of episodes by following a ε-greedy or soft-max action selection policy. In this phase, tree is hold fixed and the derived quadruples $(s_t, a_t, r_{t+1}, s_{t+1})$ are only used for Q-table update according to Q-learning update rule:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a)\right) \tag{8.5}$$

Attention control and state discretization occur in the RL-fixed phase. An important point here is that the agent only accesses the environment through its visual sensor (e.g. its CCD camera), therefore in order to determine its state, it has to traverse its saccade tree from the root node down to a leaf, which determines its state s_t at time t. In the current state, the agent performs an action according to its learned policy. At this point, based on the received reward and the next captured image, which leads to state s_{t+1}, the agent calculates perceptual aliasing for this input. After each RL-fixed phase, nodes with aliasing are detected and expanded in order to reduce aliasing. After expanding aliased states, leaf nodes without patterns in their memory are deleted. It is worth noting that memories of leaf nodes after each RL-fixed phase are deleted too. The whole process of S-Tree is shown in the algorithm one.

Table 8.1 Main algorithm for learning top-down spatial attention control and physical actions

Algorithm 1 S-Tree Main Function

1: *tree* ← create a tree with a single node
2: ***Repeat***
3: **for** i = 1 to *maxEpizodes* **do** \\ **RL- fixed phase**
4: I_t = takeImage()
5: s_t = traverseTree(*tree*, I_t)
6: [s_{t+1} r_{t+1} a_t] = selectAction(s_t)
7: Δ_t = calcDelta (s_t , a_t, s_{t+1} , r_{t+1})
8: *mem*(s_t) = gatherMem(*tree*, I_t , a_t , Δ_t)
9: **end for**
10: **for** j = 1 to $|S|$ **do** \\ **Refining aliased states**
11: **if** $|mem(s_j)| > memTreshold$ *and* checkAliasing(s_j)
12: *tree* = modifyTree(*tree*, s_j)
13: **end if**
13: **end for**
14: **for** i = 1 to *maxEpizodes* **do** \\ **Tree- fixed phase**
15: I_t = takeImage()
16: s_t = traverseTree(*tree*, I_t)
17: [s_{t+1} r_{t+1} a_t] = selectAction(s_t)
18: *Q-table* = updatePolicy(*Q-table*, s_t, s_{t+1}, r_{t+1}, a_t)
19: **end for**
20: ***Until*** (no aliasing) or (maximum iterations reached)
21: postPrune (*tree*)

Measuring Aliasing After each RL-fixed phase, algorithm refines leaves with perceptual aliasing. In order to estimate aliasing, a number of items must be accumulated under a leaf node. This is done by the agent performing some episodes running the current policy learned at the previous Tree-fixed phase. An image is captured, saccade tree is traversed in order to find the perceptual state, appropriate action is performed and a reward is received. An efficient measure of perceptual aliasing in a state (leaf node) is the TD error and can be derived from Q-learning formula as follows:

$$Q(s_t, a_t) = \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right) + Q(s_t, a_t)$$
$$= \alpha\Delta_t + Q(s_t, a_t) \quad (8.6)$$

In order to detect aliasing, all memory items under a node are clustered according to their physical actions and then if any of these clusters has a variance in Δ_t's greater than a threshold (*aliasingThreshold*), then that node has aliasing at least with respect to one action. Therefore, Δ_t reduces to (8.7), because the third term in it is the same for all clustered items under a node.

$$\Delta_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \quad (8.7)$$

Δ_t's converge to zero as the RL algorithm converges when there is no further perceptual aliasing. This is when the transition function $\mathcal{T}$ and reward function $\mathcal{R}$ are deterministic which means that source of TD error is because of misclassification. Therefore, in each step of RL, Δ_t is a measure of perceptual aliasing in a state s with respect to an action *a*. The function for detecting aliasing (checkAliasing) is shown in algorithm 2.

Table 8.2 Function for detection of aliased perceptual states

Algorithm 2 Aliasing Detector: checkAliasing(s_t)
1: **for** action $a \in$ A **do** \\ A is the set of actions
2: $mem(a)$ = all memory items with action a under s_t
3: $var(a)$ = calcVariance ($mem(a)$)
4: **if** $var(a) > aliasingThreshold$
5: return *true*;
6: **end if**
7: **end for**
8: return *false*

Tree pruning Proposed algorithm constructs an attention tree in a greedy manner which usually leads to overfitting. Therefore solutions should be designed to overcome overfitting by either periodic tree restructuring or pruning. Two heuristics are introduced in the following.

Consider two types of nodes: 1) nodes having leaves with the same best actions learned for them 2) nodes with at least one child with no memory. Leaves of the nodes in the first category are removed and their action is assigned to their parent node. A node in the second category is removed and its child with memory is substituted with it. These two heuristics are recursively done from bottom to top of the tree until no node satisfies one of these conditions. The whole algorithm for learning image-to-action mapping is summarized in the pseudocode of table 8.1.

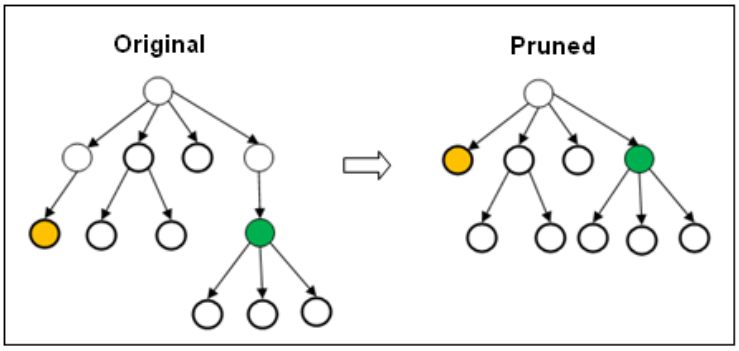


Figure 8.6 Tree pruning

Tree Refinement When an aliased class is detected, a spatial location must be selected to dissociate items under this aliased class. In order to find the best location an anticipatory mechanism is needed. When an image is classified in state s_t, codebooks in some spatial regions are saved for this node plus the Δ_t and the elicited action (gatherMem).

Assume that $M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,q}\}$ is the set of q spatial locations for the k-th leaf of the tree. Codebooks at these locations are calculated for every image which ends to this node. Let Mem_k be the matrix of memory items in the k-th leaf:

$$Mem_k = \left[mem_{i,j}\right], i = 1 \dots |Mem_k|, j = 1 \dots q + 2 \tag{8.8}$$

Each item of this memory is represented as:

$$mem_{i,j} = \left[\tau_{i,1}, \tau_{i,2}, \dots, \tau_{i,q}, a_t, \Delta_t\right] \tag{8.9}$$

where $\tau_{i,l}$ is the codebook of the l-th spatial position of i-th memory item. Note that when a new node is created in the tree, M_k is initialized in some way for this node. Two possible approaches are random selection of spatial locations or some positions relative to the end of saccade at that node (for example in some directions). Whatever method for spatial location generation must satisfy this condition that FOA's along the path from each node to the root must not be the same.

Positions are in the image coordinate frame. If some knowledge about the distribution of images in that node is available or could be gained then it might help to generate the set of spatial locations.

Whenever size of the memory under a leaf node exceeds a threshold (*memThreshold*) and it has aliasing, then tree is refined to remove aliasing. Tree refinement is then done by selecting the spatial location which mostly reduces the variance in Δ_t of memory items according to (8.10).

$$\begin{aligned}[p^* \ a^*] &= argmin_{p,a}\left(var(L) - \sum_{c=1}^{|T|}\frac{|L_{a,p,c}|}{|L_a|}var(L_{a,p,c})\right) \\ &= argmax_{p,a}\left(\sum_{c=1}^{|T|}\frac{|L_{a,p,c}|}{|L_a|}var(L_{a,p,c})\right)\end{aligned} \tag{8.10}$$

In the above formula, L is the set of Δ_t 's of all memory items, L_a is the set of Δ_t 's of items with action a. $L_{a,p,c}$ is the set of Δ_t 's of items with action a , spatial location p and codebook c. |U| and var(U) are the size and variance of a set U. p^* and a^* are the location and the action which reduces variance the most respectively. The winning spatial location p^* is saved for the node and is used for future tree traversals. Tree is expanded based on seen codebooks at location p^*. Tree modification is shown in the pseudocode of algorithm 3.

Table 8.3 Tree refinement function

Algorithm 3 Tree Refinement: modifyTree(*tree*, s_t)
1: **for** action $a \in$ A **do**
2: $mem(a)$ = all memory items with action a under s_t
3: **for** location $m \in$ M **do** \\M is the set of spatial locations
4: find p^* and a^* according to equation 9.
5: **end for**
6: **end for**

8.4 Experimental results

In order to evaluate performances of proposed algorithm, we have applied it to two visual navigation tasks which capture the main characteristics of real world scenarios. The first task is navigation in a visual grid with obstacles and a goal state and the second one in navigation in urban area. In both experiments, 5 SIFT clusters, derived from sample images from each database were used to encode the visual features.

8.4.1 Navigation in the visual girdworld

The aim in this task is to reach the goal state in the bottom of the grid marked with letter G (Figure 8.7). The agent has a set of 4 physical actions, A = {move up, move right, move left, move down} and has no access to its (x,y) position in the grid. Agent's only perception of the world is through an image of the object underneath his foot. Any movement taking the agent to an obstacle cell or outside the gird brings it a -1 punishment. When it reaches the goal state, it is rewarded a +1 signal. Each cell of the grid is carpeted with a 128×128 image of the COIL 100 object database. S-Tree has managed to recognize all the objects in an action-based manner as well as a valid policy by creating 8 distinct perceptual classes for 11 objects. Interestingly some positions with same best actions are classified under the same leaves.

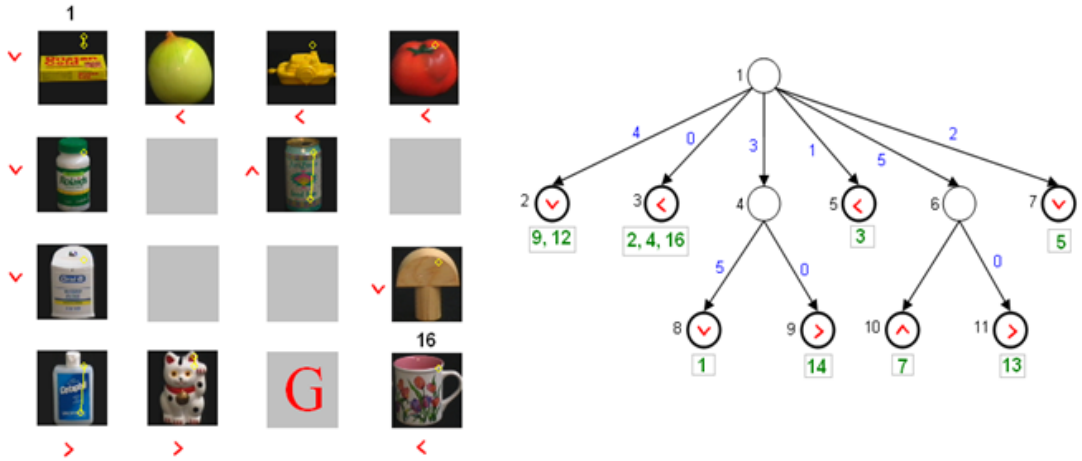


Figure 8.7 Navigation in the visual gridworld. Top: visual gridworld with learned best actions. Bottom: Learned saccade tree. Numbers below leaves indicate the positions in the grid, flashes inside the circles

show learned actions and blue numbers on the edges are the codebooks seen at the FOA. Eight spatial locations were generated in random i.e. $q=8$.

Figure 8.8 shows results of action-based object recognition and navigation in a grid with obstacles in positions 6 and 11 and goal state in position 16. In such a gird only two best actions are used from the set of four physical actions, move right and move down.

Four conditions are considered: 1) all objects for all positions are unique 2) only three positions with the same best actions are assigned the same objects 3) seven positions with the same best actions are assigned the same objects and 4) from thirteen non-obstacle, non-goal positions, seven with one action and remaining six with the other action as best are assigned the same objects. Shown by figure 8.9, as the number of distinct perceptions decrease, the number of states and hence the average tree depth also decrease. Tree pruning is shown for a grid shown in Fig 8.8. This is because there is no need to further refine the tree when there is no aliasing. Interestingly, when there are only two distinct objects in the grid (condition 4), the resultant tree has always two nodes with the average tree depth of one. The variances in diagrams are due to different initial conditions in each run like the spatial locations selected at the beginning. All cases resulted in optimal action selection policy.

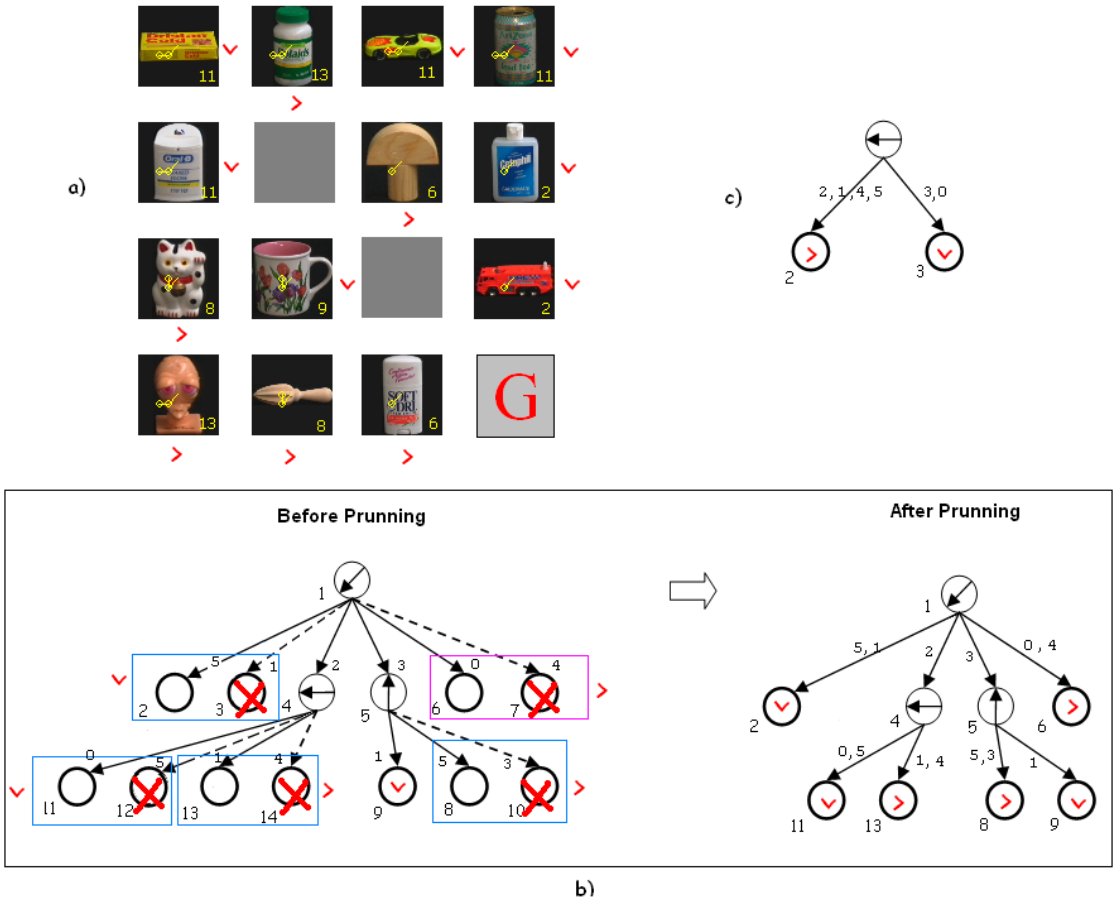


Figure 8.8 Navigation in the visual gridworld. Top: visual gridworld with learned best actions. Bottom: Learned saccade tree. Numbers below leaves indicate the positions in the grid, flashes inside the circles show

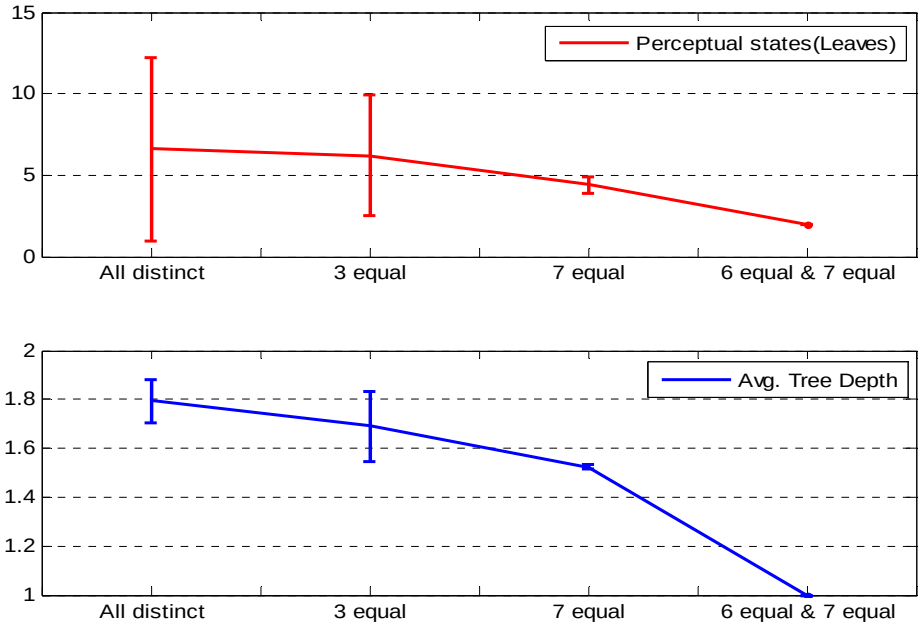


Figure 8.9 Top: Number of perceptual states Bottom: Average tree depth for a 4×4 gridworld with obstacles at positions 6 and 11 and goal at 16.

Experimenting with another more complex 10×10 gridworld shown in fig 8.10, s-tree again succeeded to derive the optimal policy after 10 phases. Number of generated visual classes was 28 equal to the number of the non-obstacle non-goal positions in the grid.

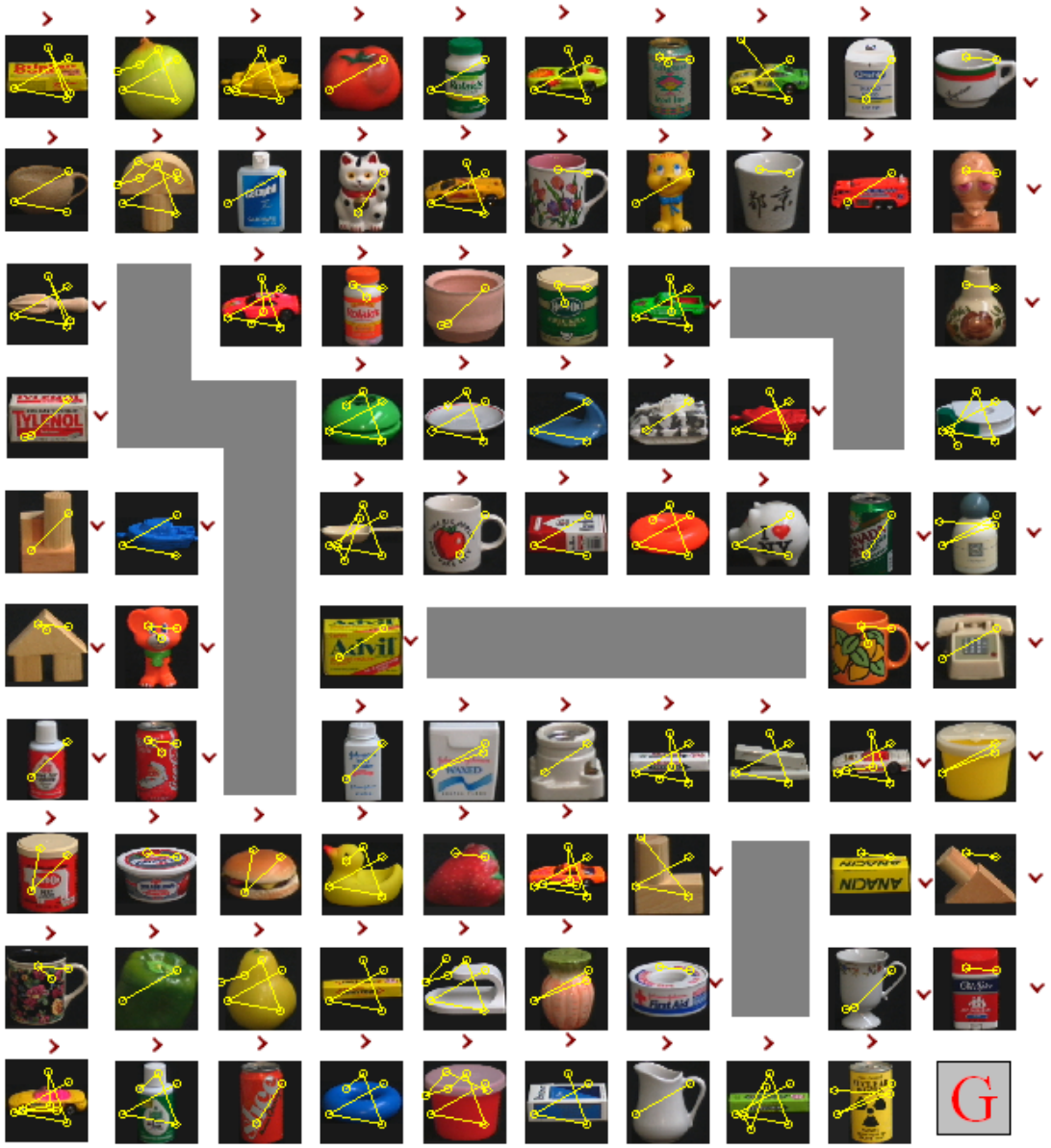
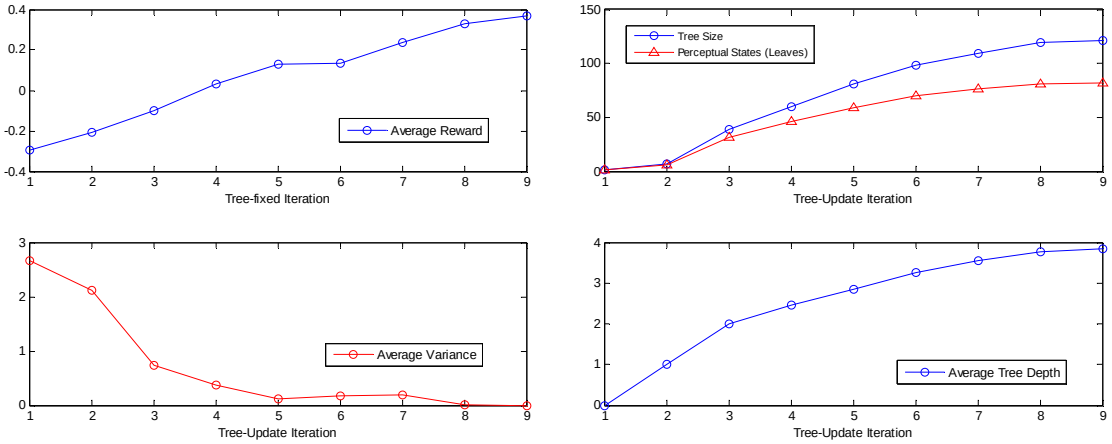


Figure 8.10 10×10 visual gridworld, maxEpizodes was 400. Other parameters were the same as in fig 5.



a) *b)*

Figure 8.11 10×10 visual gridworld, maxEpizodes was 400. Other parameters were the same as in fig 5.

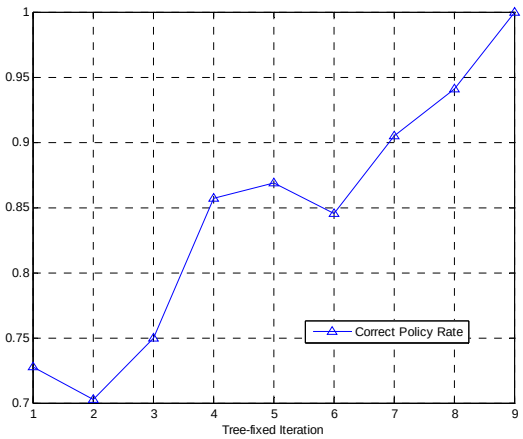
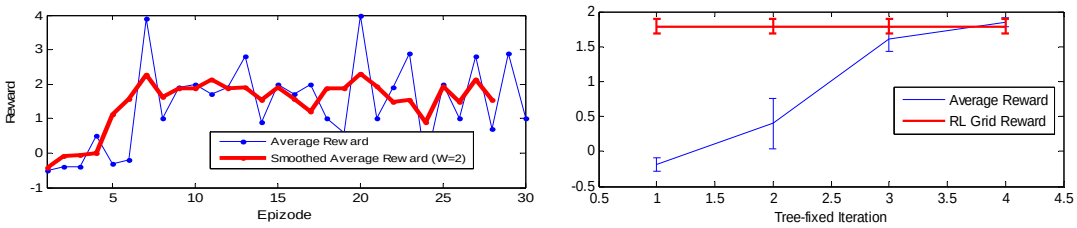


Figure 8.12 10×10 visual gridworld, maxEpizodes was 400. Other parameters were the same as in fig 5.

We also compared the traditional RL algorithm with hand designed states space with saccade tree algorithm. Here each state is the *x-y* position of the agent in the visual grid of Fig. 8.8.*a*. Fig. 8.13.*a* shows the instantaneous and smoothed average reward of the agent in 30 episodes. Agent was able to solve the task after these episodes. Figure 8.13.b shows the average reward of the agent in Tree-fixed iterations of S-Tree. Red horizontal line in this figure is the final average reward of the agent derived from Fig 8.8.a. As it can be seen both methods converged to the same average reward but traditional RL with 16 states and S-Tree with 7 states.



a) *b)*

Figure 8.13 Comparison with traditional RL

8.4.3 Handling POMDP Cases

When same objects are assigned to the positions with different best actions, algorithm does not converge. This is because there is no aliasing in perception however another kind of aliasing which emanates from not knowing the previous actions and observations exists yet. This has been predicted in original U-Tree algorithm but has not been addressed in the RLVC algorithm. It is important to note that this kind of aliasing frequently occurs in real-world situations. To remedy this, we consider a history for the agent with each item a pair of previous states and actions to a certain depth n. Let $e_{t,j}$ be the j-th pair of the history:

$$e_{t,j} = \left(a_{t-j}, s_{t-j}\right), j = 1 \dots n, \qquad a \in \{A \cup Null\}, s \in \{S \cup Null\} \tag{8.11}$$

Then new representation of a memory item becomes:

$$mem_{i,j} = \left[\tau_{i,1}, \tau_{i,2}, \dots, \tau_{i,q}, e_{t,1}, e_{t,2} \dots, e_{t,n}, a_t, \Delta_t\right] \tag{8.12}$$

History is treated the same as the spatial locations. This way when splitting a node, the history items at that node also competes for reducing aliasing. Applying this new modification, now the S-Tree algorithm is able to solve a grid with positions with different best actions having the same perceptions. In grid with obstacles at positions 6 and 11 and goal at 16, positions 2, 5, 12 and 15 were assigned the same object. As Figure 8.14 shows, S-Tree solved this problem by checking the previous action in the internal node marked with blue. Edges below this node check the previous action leaded to the current node. In the first case, S-Tree achieved to solve the task with 13 states and in the second example 9 states were created.

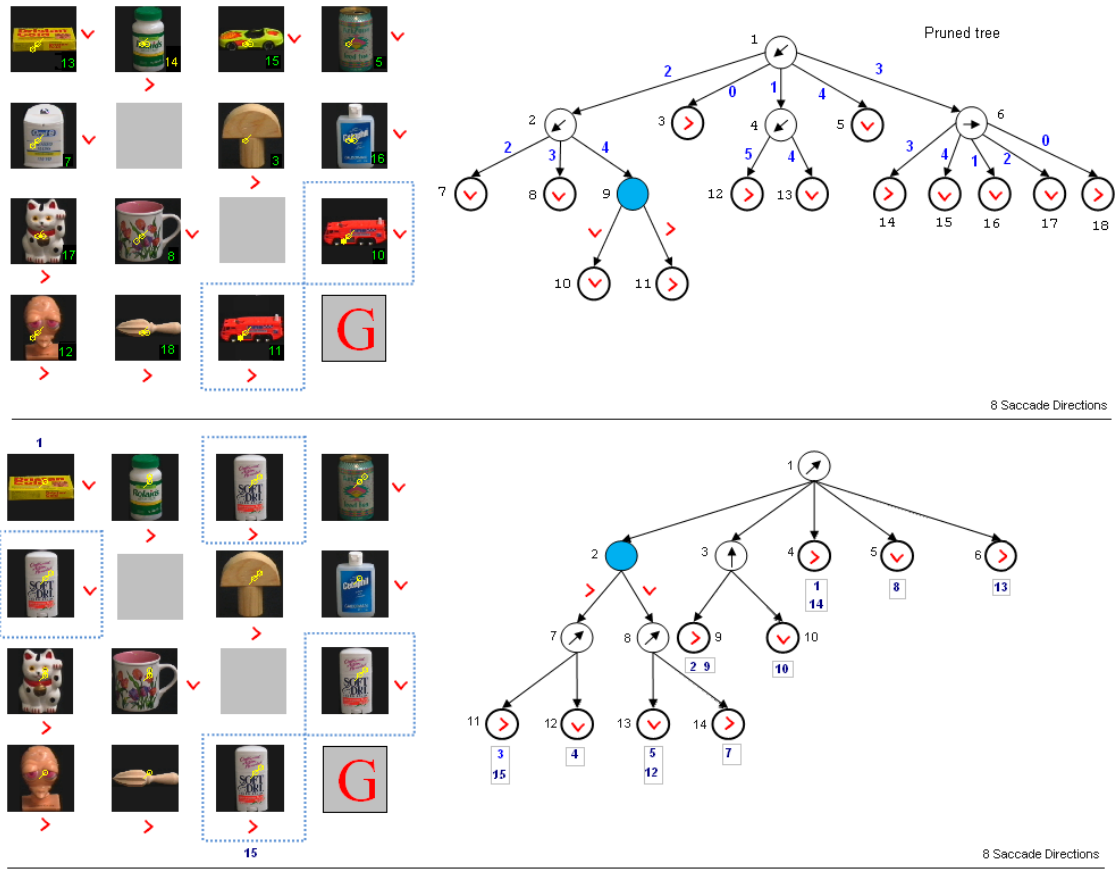


Figure 8.14 Handling POMDP cases

8.4.4 Object Recognition

Object recognition is a fundamental task of biological mechanisms. It allows an agent to abstract its knowledge and then be able to derive task independent representations. While above experiments showed that S-Tree is capable of visual navigation and action-based object classification, this experiment investigates the capability of S-Tree for classical object recognition. Basics of the algorithm are the same as before but with the distinction that here reward function is a 100×100 matrix with diagonal elements as 1 and off-diagonal values equal to -1. Class labels are assigned to the images instead of physical actions and there is no grid here. S-Tree was achieved 100% recognition rate in this case over 100 objects from the COIL100 database. Eight random spatial locations and five codebooks were used.

The only difference in algorithm is definition of the Δ_t which here reduces to only r (reward). There is no state transition here and agent each time randomly perceives or captures scenes or objects. For recognition of 16 objects the reward function is a matrix which has ones in diagonal and -1 in off diagonal elements.

$$R = \begin{bmatrix} 1 & \dots & -1 \\ \vdots & \ddots & \vdots \\ -1 & \dots & 1 \end{bmatrix}_{16\times16} \tag{8.13}$$

Fig. 8.15 shows these 16 objects along with the generated saccade tree. As it shows all objects are uniquely assigned with 16 generated leaves of the tree. The same thing could be also done for recognition of scenes. 100 object cases generated a scan path like that of Fig. 8.15.

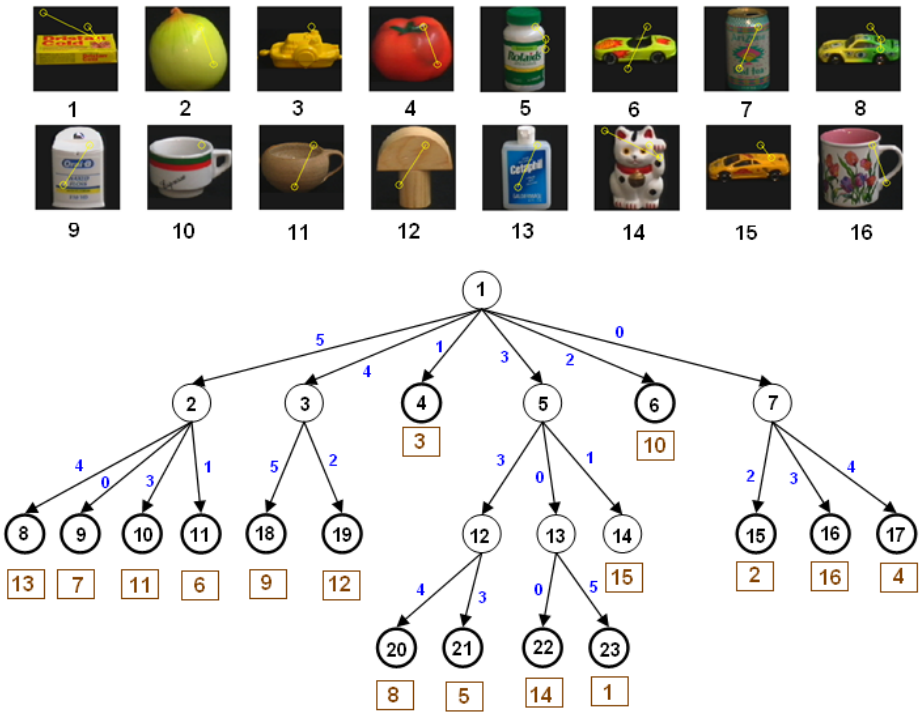


Figure 8.15 Object recognition with S-Tree

8.4.5 Invariance and Perturbation Analysis

Two types of perturbation are considered in this section 1) Perturbation of perceptual input after learning attention tree 2) Image distortions and transformations. Detailed analysis of each one is presented as follows.

Perturbation Analysis Learning happens all the time in biological creatures. It will be beneficial for a learning system to be able to adapt itself to dynamics of the environment and perceptual space. In the first experiment in this section, we altered the image at position 15 in the grid used in Fig. 8.8.*a* to a new unseen image after learning and then learning was continued. S-Tree expanded node 9 in the gird and created three new states instead as shown by Figure 8.16. Pruning removed node 4 which classified image at position 15 earlier.

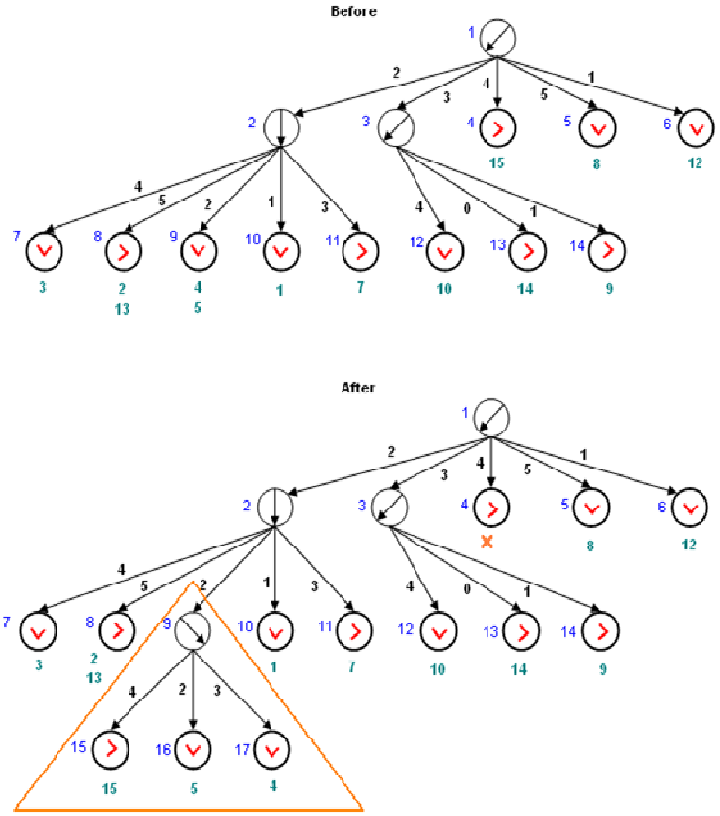


Figure 8.16 When After learning three positions (4 8 12) have made same images.

In the second experiment, after learning took place in iteration three images at locations 5 and 8 were randomly changed. Algorithm converged again we changed the image at location 7 and it converged again. Correct policy rate of this event is shown in Fig 8.17.

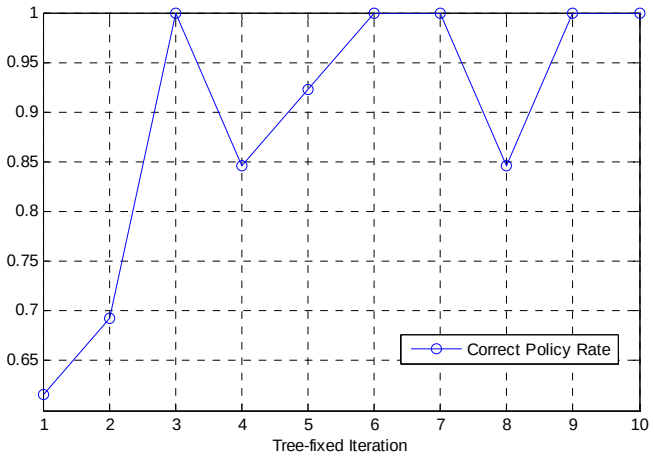


Figure 8.17 Perturbation of perceptual inputs after learning

This result shows that S-Tree is to some extent robust to distortions and perturbations in perceptual inputs.

Invariance Analysis Previous experiments showed that S-Tree achieved 100% correct policy rate or object recognition rate when only a single image per class or position was used. However, this is not a right assumption for the perceptual space. A learning system while being able to discriminate among samples from different classes (specificity) has to be able to treat all the samples within a class equally (generalization). In this experiment we investigate generalization power of the S-Tree algorithm.

A gridworld with obstacles at positions 6 and 11 and the goal at 16 is used. Each position of this grid is assigned a natural 640×480 scene[4]. In each position, the agent captures a scene randomly among 5 possible scenes with major transformation (i.e it does not observe the same image per position but observes images from the same scene under major translation, scale and rotation). The goal here is to learn the correct policy but with minimum number of states or leaves. In all cases agent was able to solve the task. To measure the generalization power an index know as *state reduction index* is defined as in (8):

$$SRI = \frac{1}{N}\sum_{n=1}^{N}\frac{\sigma^2\left(V(n)\right)}{mean\left(V(n)\right)},$$
$$V(n) = [v_1(n), v_2(n), \dots, v_5(n)] \qquad (8.14)$$

In above formula, $V(n)$ is the vector of leaf numbers and N is tshe number of objects. Numbering is done at the level order when tree is built. Zero value for *SRI* means that all samples of a scene are classified under the same leaf. Lower the *SRI*, means higher generalization power. Figure 8.18.a compares the *SRI*, number of states and average tree depth (average of depths of all leaves) for two SIFT selection methods in S-Tree (f_1 and f_2) and RLVC. As can be seen highest magnitude SIFT is more invariant to image transformations than the nearest SIFT feature. RLVC has more generalization because it checks the existence of a SIFT feature anywhere in the image but with the disadvantage of more computation. Lower *SRI* means that lower number of states in average as Figure 8.18.a shows. Average tree depth of three approaches is nearly the same.

Figure 8.18.b shows average number of leaves, number of tree nodes and phases (turns) after convergence of the S-Tree for recognition of 16 scenes (5 views each), where class labels are used instead of actions. Consistent with 6.a, it shows that largest magnitude SIFT leads to more generalization. Increasing the number of codebooks leads to lower average tree depth as shown by Figure 8.18.c.

[4]Data for this experiment was extracted from: http://www.montefiore.ulg.ac.be/~jodogne/phd-database.

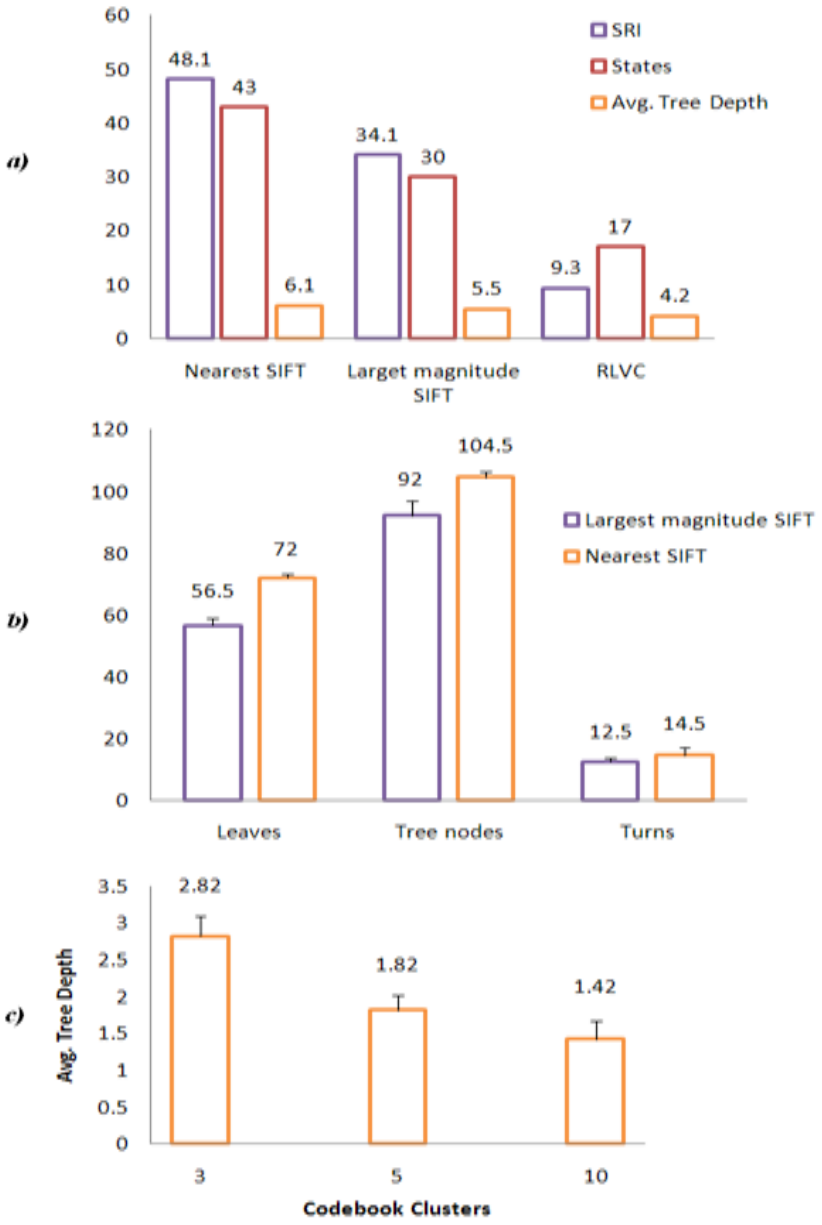


Figure 8.18 Invariance analysis a) comparing S-Tree and RLVC over a visual navigation task with r = 50 and 5 codebooks, b) object recognition task and c) effect of number of codebook clusters on average tree depth over a 4×4 grid with random spatial locations and one view per position.

8.5 Discussions

We used the idea of state-space discretization, to emulate the need-based mechanism that human brain has adopted for representation and attention. The mechanism of growing the saccade tree is to refine those tree leaves with aliasing by selecting a spatial location among a set of locations which reduces aliasing the most according to equation (8.9).

Results show that S-Tree is able to solve the gridworld and object recognition tasks very fast by only extracting SIFT features at small number of image regions. Compared with RLVC which has $O(kn^2)$ computational complexity where k is the average tree depth and n is the image size,

complexity of S-Tree is $O(k'\pi r^2)$ where the average tree depth or saccade length is k' and r is the radius of FOA. Ignoring the constants, S-Tree is n/r times more faster than RLVC. Since, this is quite appealing, a shortcoming arises from this local processing and it is lack of generalization. Since S-Tree algorithm works in spatial domain instead of feature space, it makes the method sensitive to large spatial transformations like translation, rotation and scale. RLVC is more robust to these transformations but with the expense of high computation and the fact that its behavior seems not to be much correlated with human attentional behavior.

8.6 Conclusions

The main contribution of our algorithm is generating saccade trees which only needs calculating the SIFT features at few spatial locations of the image. It is also in accordance with some behavioral attentional mechanisms and is categorized under the space-based theories of visual attention. Actually saliency of the image locations is determined by the expected reward they convey in an interactive task.

In this study we avoided using detailed complex camera like representation for world and instead incorporated attention, observation and motor action for learning internal representations and attentions. Our findings partially point toward the fact that saccadic eye movements and active perception control, together with active movements may play a crucial role in the mechanisms that the brain uses to represent the world.

An interesting observation is that representations are learned interactively and are expanded adaptively based on the agent's needs. They are also as compact as possible and encode the information at the necessary level without unnecessary details. For example, for recognition of a scene, it would be very efficient and conclusive to attend to important spatial locations. Therefore, global image representation approaches although might propose more accurate solutions in some cases seems not to be the best solutions where information bottlenecks exit.

In accordance with these views, our method discretizes the visual world when it is needed and when it helps the agent to perform better by removing perceptual aliasing. The only predefined knowledge supplied to the agent was the clusters of visual features or codebooks. This does not put a big constraint on the method because these clusters could also be learned.

The weakness of S-Tree in generalization is because saccadic movements are initiated in a coordinate frame locked to the image. This causes relocation of visual contents at FOA when an image is transformed. While we tried to remedy this by considering the codebook of the highest magnitude SIFT to some extent, problem still remains to be investigated in future researches. A possible solution is by introducing a coordinate frame which is relative to a stable property of the image. For example a set of stable landmarks which could be calculated very fast in a pre-attentive manner would be very desirable. Salient points introduced by the bottom-up saliency based model of visual attention (Itti, et al., 1998) could be promising candidates.

Chapter 9
Conclusions and Perspectives

This chapter gives a summary of the results acquired in this thesis, accompanied by some suggestions for future researches. The goal of the thesis was to develop solutions for learning top-down task-driven visual attention control in visual interactive environments.

9.1 Summary of the Contributions

In our first contribution we were motivated by learning associations between situations to attentions. For example drivers often attend to cross light to update their status. These associations are present in a large number of our behaviors. For learning such associations we used RL, and the reward punishments were given by a critic whenever agent performed the right association. However, learning here seems to be supervised than semi-supervised. This type of solution is not in contradiction to our other solutions presented in this thesis. One justification could be that these associations could be results of higher abstractions of lower-level behaviors which could be learned by RL.

Our second contribution deals with biasing the bottom-up saliency based model of visual attention. The key idea here is that some channels and scales of the saliency model are less important for detection of an object. To this end, we designed a new center-surround operation for making the scales of the model independent of each other. Instead of finding a closed –form solution for optimal object detection, we formulated the object detection problem as an optimization problem and solved it by global optimization algorithm. The proposed solution was able to efficiently in terms of computational complexity to detect synthetic saliencies as well as natural objects in cluttered scenes. This forms the basis for our third contribution which is a top-down object-based attention control mechanism. This model consists of three layers. First, in the early visual processing layer, basic layout and gist of a scene are extracted. The most salient location of the scene is simultaneously derived using the biased saliency-based bottom-up model of visual attention. Then a cognitive component in the higher visual processing layer performs an application specific operation such as object recognition and scene understanding at the focus of attention. From this information, a state is derived in the decision making layer. Top-down attention in our model is learned by the U-TREE algorithm which successively grows a tree whenever perceptual aliasing occurs. Internal nodes in this tree check the existence of a specific object in the scene and its leaves point to states in the Q-table. Motor actions are associated with leaves. After performing a motor action, the agent receives a reinforcement signal from the critic. This signal is alternately used for modifying the tree or updating the action selection policy.

In this contribution an approach for scene classification was proposed which extracts and matches visual features only at the focuses of visual attention instead of the entire scene. Analysis over a database of natural scenes demonstrates that regions proposed by the saliency-based model of visual attention are robust with image transformations. Classification results show that classification rate is nearly the same as when features are extracted over the entire scene but with feature extraction at a small fraction of scene regions. An approach for simultaneous learning of physical actions and representations known as RLVC is also extended by limiting it's SIFT search and extraction to only salient regions. Results prove that RLVC has still the same performance with a huge decrease in computational complexity. Overall, our results prove that efficient scene classification, in terms of reducing the complexity of feature extraction is possible without a significant drop in performance.

The fifth contribution introduces an approach for simultaneous and interactive learning of space-based visual attention control and physical actions. Our approach is based on the RLVC algorithm and adapts it for learning spatial visual selection in order to reduce computational complexity. Proposed algorithm also addresses aliasings due to not knowing previous actions and perceptions. Continuing learning shows our method is robust to perturbations in perceptual information. Proposed method also allows object recognition when class labels are used instead of physical actions. We have tried to gain maximum generalization while performing local processing. Experiments over visual navigation and object recognition tasks show that our method is more efficient in terms of computational complexity and is biologically more plausible.

9.2 Future Work

Human visual learning and visual attention is very complex task especially we are dealing with natural scenes. While much is known on visual structures and functionalities in the brain, there is still very little known how these functionalities are formed through development or evolution. This is very interesting topic that deserves much research focus. Despite our efforts in this thesis, there are still problems in these areas that should be tackled. In this section, we mention some of those problems along with other interesting directions for future studies.

Remedy the lack of Generalization

Throughout this thesis, especially last chapters, we were looking for solutions for processing subsets of spatial regions. We wanted to know how spatial attention mechanisms could be learned. This is motivated by the spatial attention behavior of humans. The problem in the saccade tree is its low generalization power. This arises from the fixed coordinate frame on the image. For example our agent supposes to see the same feature at the same spatial location over image an transformation which is not true in general. The remedy to this could be using a relative reference frame instead of an absolute coordinate frame. One solution could be using the salient points to define such a relative frame. Another solution could be employing statistical approaches like Bayesian Reinforcement Learning (RL) or Bayesian Belief Networks (BBN). As another solution, could be attending to objects rather spaces or a combination of both which seems to be the case in humans. Humans attend to both spaces and objects at different situations. Attending to objects has this benefit that objects bring the generalization provided that an object based solution

is capable of detecting that object in any of its transformations. Whatever solution for this problem should have the property of reducing the visual processing while solving the task.

Modeling the Effect of Context and Spatial relationships

Spatial context has a strong impact on direction of human attention. For example when looking for a clock, we simply are oriented toward the walls. It seems that we have a model or memory of the visual environment we live and work in. It is less known how these spatial relationships can be defined and how they facilitate visual attention. These relations might be essential for spatial attention and may help solve the generalization problem. Effect of context could be implemented in the saccade tree by directing the search to a subset of subtrees.

Scaling up the Solutions

Our experiments were designed just to prove the functionality of proposed approaches. It would be worth to apply them to larger datasets and cluttered scenes. An interesting direction could be applying the s-tree to a dataset of natural scenes in which objects are tagged (Russell et al. 2008 IJCV). This database is becoming a benchmark for testing the proposed solutions for tackling the complex problems like attention which could not be studied unless applying them to large scale data.

Implementation on real robots

Neither we nor Jodogne & Piater (2007), have applied their solutions on real robots dealing with natural scenes and complexities of online working with a cluttered environment. One disappointing reason stems from this weakness of RL which needs a large number of training episodes which usually needs a lot time. In this regard, it would be rewarding to extend the purposive and need-based solutions for visual learning for embodied robots. Current solutions does not assume a body for agents. It is evident that body form has a great impact on visual attention and visual behaviors. For example different animals have different visual capabilities due to different tasks they do and their bodies.

Bibliography

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

Z. Li, A saliency map in primary visual cortex. Trends in Cognitive Sciences. 6, 9-16 (2002)

M. Corbetta, G.L. Shulman, Control of goal-directed and stimulus-driven attention in the brain. Nature Reviews. 3, 201-215 (2002)

M.I. Posner, Orienting of attention. Quarterly Journal of Experimental Psychology. 32, 3-25 (1980)

M.I. Posner, Y. Cohen, Components of visual orienting. In: Attention and Performance X, edited by Bouma H and Bouwhuis D. Hillsdale: Erlbaum, 531-556 (1984)

J.H. Maunsell, S. Treue, Feature-based attention in visual cortex, Trends in Neurosciences. 29, 317-322. TINS special issue: The Neural Substrates of Cognition (2006)

N. Kanwisher, J. Driver, Objects, attributes, and visual attention: which, what, and where. Current Directions in Psychological Science. 1, 26-31 (1992)

A.L.Yarbus, Eye movements during perception of complex objects, in L. A. Riggs, ed., Eye Movements and Vision, Plenum Press, New York, chapter VII. 171-196 (1967)

L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. 20, 1254-1259 (1998)

K. Nakayama, V. Maljkovic, A.Kristjansson, Short term memory for the rapid deployment of visual attention. M.S. Gazzaniga (Eds.), The Cognitive Neurosciences, 3rd edition. Cambridge, MA, USA: MIT Press. (2004)

K. Nakayama, M. Mackeben, Sustained and transient components of focal visual attention. Vision Research. 29, 1631-1647 (1989)

A. Kristjansson, K. Nakayama, A primitive memory system for the deployment of transient attention. Percept. Psychophys. 65, 711-724 (2003)

V. Maljkovic, K. Nakayama, Priming of popout I. role of features. Mem. Cogn. 22, 657-672 (1994)

I. A. Rybak, V. I. Gusakova, A.V. Golovan, L.N. Podladchikova, , Shevtsova, N.A.: A model of attention-guided visual perception and recognition. Vision Research. 38, 2387-2400 (1998)

R.M. Klein, Inhibition of return. Trends in Cognitive Sciences. 4, 138-147 (2000)

C. Koch, S. Ullman, Shifts in selective visual attention: towards the underlying neural circuitry. Human Neurobiology. 4, 219-227 (1985)

R. J. Peters, L. Itti, Applying computational tools to predict gaze direction in interactive visual environments. ACM Transactions on Applied Perception. 5(2), Article 8, 2008.

V. Navalpakkam, L. Itti, Modeling the influence of task on attention. Vision Research. 45, 205-231 (2005)

S. Frintrop, VOCUS: A visual attention system for object detection and goal-directed search. PhD thesis, Lecture Notes in Artificial Intelligence (LNAI). 3899 (2006)

V. Navalpakkam, L. Itti, An integrated model of top-down and bottom-up attention for optimizing detection speed. Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2, 2049-2056 (2006)

D.M. Gavrila, Traffic sign recognition revisited. In Mustererkennung (DAGM), Bonn, Germany, Springer Verlag. (1999)

N. Barnes, A. Zelinsky, Real-time radial symmetry for speed sign detection. In IEEE Intelligent Vehicles Symposium (IV), pages 566-571, Parma, Italy. (2004)

A. de la Escalera, J.M. Armingol, M. Mata, Traffic sign recognition and analysis for intelligent vehicles. Image and Vision Computing. 21, 247-258 (2003)

P. Paclik, J. Novovicova, P. Somol, P. Pudil, Road sign classification using Laplace kernel classifier. Pattern Recognition Lett. 21(13–14), 1165-1173 (2000)
L.C. Gomez, O. Fuentes, Color-based road sign detection and tracking. Proceedings Image Analysis and Recognition (ICIAR), Montreal, CA. (2007)
C.Y. Fang, S.W. Chen, C.S. Fuh, Road-sign detection and tracking. IEEE Trans. Vehicular Technology. 52(5), 1329-1341 (2003)
A. de la Escalera, L. Moreno, Road traffic sign detection and classification. IEEE Trans. Indust. Electronics. 44, 848-859 (1997)
P.J. Burt, E.H. Adelson, The Laplacian pyramid as a compact image code. IEEE Transactions on Communications. 31(4):532-540 (1983)
L. Itti, C. Koch, Feature combination strategies for saliency-based visual attention systems. Journal of Electronic Imaging. 10(1), 161-169 (2001)
Saliency toolbox homepage. http://www.saliencytoolbox.net/
J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evolutionary computation. 10(3), 281-295 (2006)
J.M. Wolfe, Visual search. In H. Pashler (Ed.), Attention, East Sussex, UK: Psychology Press. (1998)
H.E. Egeth, S. Yantis, Visual attention: control, representation, and time course. Annual Review of Psychology, 48(1997) 269–297.
C.E. Connor, H.E. Egeth, S. Yantis, Visual attention: Bottom-up versus top-down, Current Biology, 14(2004) 850-852.
M. Corbetta, G. L. Shulman, Control of goal-directed and stimulus-driven attention in the brain. Nature Reviews, 3(2002) 201–215.
R. Desimone, J. Duncan, J., Neural mechanisms of selective visual attention Annual Review of Neuroscience, 18(1995) 193-222.
M. M. Chun, J. M. Wolfe, Visual Attention. In E. B. Goldstein (Ed.), Blackwell's Handbook of Perception, 9 (2001) 272-310). Oxford, UK: Blackwell.
C. Koch and S. Ullman, Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry, Human Neurobiology, 4(1985) 219–227.
Z. Li, A saliency map in primary visual cortex Trends in Cognitive Sciences 6(2002): 9-16.
S. Kastner, L. G. Ungerleider, The neural basis of biased competition in human visual cortex. Neuropsychologia. 39(2001) 1263–1276.
M. I. Posner, Orienting of attention. Quarterly Journal of Experimental Psychology, 32(1980) 3–25.
M.I. Posner, Y. Cohen. Components of visual orienting. In: Attention and Performance X, edited by Bouma H and Bouwhuis D. Hillsdale: Erlbaum, (1984) 531–556.
J. H. Maunsell, S. Treue, Feature-based attention in visual cortex, trends in neurosciences, 29 (2006) 317-322, TINS special issue: The Neural Substrates of Cognition, 2006.
N. Kanwisher, J. Driver, Objects, attributes, and visual attention: Which, what, and where. Current directions in Psychological Science, 1(1992) 26-31.
J. H. Duncan, Selective attention and the organization of visual information Journal of Experimental Psychology: General, 113(1984) 501-517.
D. Kahneman, A. Henik, Perceptual organization and attention. In M. Kubovy & J. R. Pomerantz (Eds.), Perceptual organization (1981) 181-211. Hillsdale, NJ: Erlbaum.
A. L. Yarbus, Eye movements during perception of complex objects, in L. A. Riggs, ed., Eye Movements and Vision', Plenum Press, New York, chapter VII ((1967) 171- 196.
V. Maljkovic, K. Nakayama, Priming of pop-out: I. Role of features. Mem. & Cognition, 22(1994) 657-672.
E. Gibson, E. Spelke The development of perception, Handbook of child psychology vol. iii: cognitive development, chapter, 1, 2-76, Wiley. 1983.

M. J. Tarr,Y.D. Cheng. Learning to see faces and objects. Trends in Cognitive Sciences, 7(2003) 23–30.
W. D. Gray, (Ed.), Integrated models of cognitive systems. New York: Oxford University Press, (2007).
A. Clark, and R. Grush, Towards a cognitive robotics. Adaptive Behavior, 7(1999).
R. Pfeifer and J. C. Bongard , How the Body Shapes the Way We Think A New View of Intelligence, MIT press, 2006.
A. Clark, Where brain, body, and world collide, Journal of Cognitive Systems Research, 1(1999) 5–17.
J. Triesch, D. H. Ballard, M. M. Hayhoe, B. T. Sullivan, What you see is what you need. Journal of Vision, 3(2003), 86–94.
R. S. Sutton, A. G. Barto, Reinforcement Learning. The MIT Press. Cambridge, MA, 1998.
B. Seymour, J.P. O'Doherty, P. Dayan, K. Koltzenburg, A.K. Jones, R.J. Dolan, K.J. Friston, R.S. Frackowiak, Temporal difference models describe higher order learning in humans, Nature. 429(2004) 664-667. doi:10.1038/nature02636.
L. Itti, C. Koch, E. Niebur, A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1998) 1254-1259.
L. Itti, C. Koch, Computational Modeling of Visual Attention. Nature Reviews Neuroscience, 2 (2001) 195–203.
V. Navalpakkam, L. Itti, Modeling the influence of task on attention. Vision Research, 45 (2005) 205–231.
V. Navalpakkam, L. Itti, An Integrated Model of Top-Down and Bottom-Up Attention for Optimizing Detection Speed, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference, 2(2006) 2049 – 2056.
A. Torralba, Modeling global scene factors in attention, Journal of Optical Society of America A. Special Issue on Bayesian and Statistical Approaches to Vision. 20(2003): 1407-1418.
A.K. McCallum, Reinforcement learning with selective perception and hidden state. Doctoral dissertation, Department of Computer Science, University of Rochester. 1995.
D. Gabor, Theory of Communication, Journal of the Institute of Electrical Engineers, 93(1946), 429-457.
D. Hubel, T. Wiesel, Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat .J.Neurophys 28(1965) 229-289.
M. C. Mozer, M. Shettel, S. P.Vecera, Control of visual attention: A rational account. In Y. Weiss, B. Schoelkopf, & J. Platt (Eds.), Neural Information Processing Systems 18(2006) 923-930. Cambridge, MA: MIT Press.
E. D. Reichle, L. Patryk. A., Using Reinforcement Learning to Understand the Emergence of Intelligent Eye-Movement Behavior During Reading, Psychological Review Copyright 2006 by the American Psychological Association 113 (2006) 390- 408, 2006.
N. Sprague, D. H. Ballard, Modeling Embodied Visual Behaviors. 2005.
G. Fritz, C. Seifert, L. Paletta, H. Bischof, Attentive Object Detection Using an Information Theoretic Saliency Measure. WAPCV 2004: 29-41
L. Paletta, G. Fritz, C. Seifert, Cascaded Sequential Attention for Object Recognition with Informative Local Descriptors and Q-learning of Grouping Strategies , Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
L. M. G. Gonic, G. A. Giraldi, A. AF. Oliveira, P.A. Grupen, Learning Policies for Attentional Control, IEEE International Symposium on Computational Intelligence in Robotics and Automation, (1999) 294 - 299.

L. M. G. Gonic, A. Antonio, A. AF. Oliveira, and P.A. Grupen, A Framework for Attention and Object Categorization Using a Stereo Head Robot, XII Brazilian Symposium on Computer Graphics and Image Processing, Proceedings (1999) 143-152.
S. Minut, S. Mahadevan, A Reinforcement Learning Model of Selective Visual Attention", Fifth International Conference on Autonomous Agents, Montreal, 2001.
H. F. Shariatpanahi H. F., M. N. Ahmadabadi, Biologically Inspired Framework for Learning and Abstract Representation of Attention Control, Proceedings of International Workshop on Attention in Cognitive Systems, at IJCAI 2007, Hyderabad, India, (2007) 63-80.
R. J. Peters, L. Itti, Applying computational tools to predict gaze direction in interactive visual environments, ACM Transactions on Applied Perception, Vol. 5, No. 2, p. Article 8, 2008.
D. Walther, U. Rutishauser, C. Koch, and P. Perona, Selective visual attention enables learning and recognition of multiple objects in cluttered scenes, Computer Vision and Image Understanding (2005), 100, 41-63.
D. Lowe, Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 60(2004), 2, 91–110.
D. Walther and C. Koch, Modeling attention to salient proto-objects. Neural Networks 19(2006) 1395-1407.
R. Egly, J. Driver, R.D. Rafal, Shifting visual attention between objects and locations: Evidence from normal and parietal lesion subjects. Journal of Experimental Psychology General, 123(1994), 2, 161–177.
M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex. Nature Neuroscience, 2(1999),11, 1019–1025.
M. Riesenhuber, T. Poggio, How visual cortex recognizes objects: the tale of the standard model. In L. M. Chapula, & J. S.Werner (Eds.), The visual neurosciences (pp. 1640–1653). Cambridge, MA: MIT Press (2003).
S. Jodogne, J. H. Piater, Closed-Loop Learning of Visual Control Policies. Journal of Artificial Intelligence Research 28(2007) 349–391.
G. Deco, B. Schürmann, A hierarchical neural system with attentional top-down enhancement of the spatial resolution for object recognition. Vision Research, 40(2000),20, 2845–2859.
J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. H. Lai, N. Davis, F. Nuflo, Modeling visual-attention via selective tuning. Artificial Intelligence, 78(1995) 507–545.
F. H. Hamker, The reentry hypothesis: The putative interaction of the frontal eye field, ventrolateral prefrontal cortex, and areas V4, IT for attention and eye movement. Cerebral Cortex, 15(2005), 4, 431–447.
F. H. Hamker, The emergence of attention by population-based inference and its role in distributed processing and cognitive control of vision. Computer Vision and Image Understanding, 100(2005), 1–2, 64–106.
Y. Sun, Y., R. Fisher, Object-based visual attention for computer vision. Artificial Intelligence, 20(2003), 11, 77–123.
A. Oliva, A. Torralba, Building the Gist of a Scene: The Role of Global Image Features in Recognition. Progress in Brain Research: Visual perception, 155(2006) 23-36.
P. Burt, E. Adelson, The Laplacian Pyramid as a compact image code, IEEE Trans. Commun. COM-31 4(1983)532– 540.
E. Simoncelli, W. Freeman, The steerable pyramid: a flexible architecture for multi-scale derivative computation, in: Internat. Conf. on Image Processing, 1995.
L. Itti, C. Koch, Feature combination strategies for saliency-based visual attention systems. Journal of Electronic Imaging, 10(2001), 1, 161–169.
J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE trans. evolutionary computation, 9(2006), 3.

C. Siagian and L. Itti, Rapid biologically-inspired scene classification using features shared with visual attention, IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 29, no. 2, pp. 300–312, Feb 2007
L. Renniger and J. Malik, When is scene identification just texture recognition? Vision Research, 44(2004) 2301–2311.
A. Torralba, K. P. Murphy, W. T. Freeman, M. A. Rubin, Context-based vision system for place and object recognition, in IEEE Intl. Conference on Computer Vision (ICCV), Nice, France, (2003) 1023 – 1029.
C. Siagian, L. Itti, Comparison of gist models in rapid scene categorization tasks, In: Proc. Vision Science Society Annual Meeting (VSS08), May 2008.
P. Vuilleumier, How brains beware: neural mechanisms of emotional attention, TRENDS in Cognitive Sciences, 9(2005), 12.
T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, Object recognition with cortex like mechanisms, IEEE Trans. Pattern Anal. Machine Intell. 29(2007), 3, 411-426.
T. Serre, L. Wolf, T. Poggio, 2004. A new biologically motivated framework for robust object recognition. Technical Report CBCL Paper 243/AI Memo 2004- 026, Massachusetts Institute of Technology, Cambridge, MA. (2004).
T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, T. and Poggio, A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex, AI Memo 2005-036/CBCL Memo 259, Massachusetts Inst. Of Technology, Cambridge, MA. (2005).
A. Borji, M. Hamidi, F. Mahmoudi, Robust Handwritten Character Recognition with Features Inspired by Visual Ventral Stream, Neural Processing Letters, 2008, 8(2008), 2, 97-111.
V. N. Vapnik. The Nature of Statistical Learning Theory, Springer-Verlag, NewYork, 1995.
C. Watkins and P. Dayan, Q-learning, Machine Learning, 8(1995), 3, 279–292.
J. R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
B. Russell, A. Torralba, K. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation, International Journal of Computer Vision, 77(2008) 157-173.
J. M. Wolfe, Visual Search. In H. Pashler (Ed.), Attention, East Sussex, UK: Psychology Press. 1998.
D. B. Walther and C. Koch, Attention in Hierarchical Models of Object Recognition. in Paul Cisek, Trevor Drew, and John F. Kalaska (eds.), Computational Neuroscience: Theoretical insights into brain function, Progress in Brain Research, 165(2007), 57-78.
M. Asadpour, M. N. Ahmadabadi, and R. Siegwart. Reduction of learning time for robots using automatic state abstraction. In H.I. Christensen, editor, Proc. of the First European Symposium on Robotics, volume 22 of Springer Tracts in Advanced Robotics, Palermo, Italy, Springer-Verlag, (2006)79-92.
A. Borji, M. N. Ahmadabadi, Babak N. Araabi, Learning top-down feature based attention control, Workshop on 'Vision in Action: Efficient strategies for cognitive agents in complex environments', LNCS, ECCV 2008.
A. Borji, M. N. Ahmadabadi, Babak N. Araabi, Interactive Learning of Top-down Attention Control and Motor Actions, Workshop on From motor to interaction learning in robots, LNCS, IROS 2008.
M. Boutell, C. Brown, and J. Luo, “Review of the State of the Art in Semantic Scene Classification,” Univ. Rochester, 2002.
J. G. Daugman, “Two-Dimensional Spectral Analysis of Cortical Receptive Field Profile,” Vision Res., 1980.
R. Epstein, A. Harris, D. Stanley, and N. Kanwisher, “The Parahippocampal Place Area: Perception, Encoding, or Memory Retrieval?” Neuron, 2000.
A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis, Wiley, 2001.

C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” Alvey Vision Conference, 1988.
H. Hotelling, “Analysis of A Complex of Statistical Variables into Principal Components,” J. of Educational Psy., 1933.
X. He and P. Niyogi, “Locality Preserving Projections,” NIPS.
L. Itti, C. Koch and E. Niebur, “A Model of Saliency-based Visual Attention for Rapid Scene Analysis,” IEEE T-PAMI, vol. 20, pp. 1254-1259, 1998.
D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” IJCV, vol. 60, pp. 91-110, 2004.
S. Lazebnik, C. Schmid, and J. Ponce, “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories,” IEEE CVPR, 2006.
A. Oliva, A. Torralba, Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope, IJCV, vol. 42, no. 3,pp. 145-175, 2001.
T. Poggio and E. Bizzi, “Generalization in Vision and Motor Control,” Nature, pp. 768-774, 2004.
C. Siagian and L. Itti, “Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention,” IEEE T-PAMI, pp. 300-312, 2007.
T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Object Recognition with Cortex-like mechanisms,” IEEE T-PAMI, 2007.
X. Li, et al., "Discriminant Locally Linear Embedding with High Order Tensor Data," IEEE T-SMC-B, 2008.

تقدیم به :

پدرم و مادرم

چکیده

یکی از خصوصیات بارز انسانها کارا بودن آنها در محیطهایی است که اطلاعات حسی زیادی دریافت می کنند. بینایی مهمترین حسی است که انسانها بر آن تکیه دارند و به همین علت است که این حس بیشترین مطالعات را در بینایی ماشین و هوش مصنوعی به خود اختصاص داده است. علیرغم تحقیقات وسیع در بینایی ماشین و رباتیک تعداد زیادی از اعمال حسی – حرکتی که در انسانها به سادگی انجام می دهند ، هنوز حل نشده اند. بطور خاص طراحی الگوریتمهای یادگیری که دارای دقت بالا و پیچیدگی محاسباتی پایین باشند و رباتهای متحرک خودکار را قادر سازند تا در محیطهای تعاملی بینایی عمل کنند بسیار مورد علاقه است. در مقایسه با محیط های بینایی کنترل شده که اغلب در آزمایشگاه استفاده می شوند، یادگیری رفتارهای بینایی در محیط های کنترل نشده و بصورت کلی بسیار مشکل ترمی باشد. نمونه های از کاربردهای یادگیری بینایی ، راهبری مبتنی بر بینایی ، تشخیص محل بر اساس اطلاعات بینایی ، گرفتن و حرکت دادن اشیاء می باشند.

گرایش اخیر در رباتیک به سمت توسعه ربات هایی است که قادر باشند بصورت خودکار در محیط های بینایی ناآشنا و تصادفی عمل کنند. این کیفیت مورد علاقه ، روشهای بر خط و تعاملی را برای یادگیری نمایشهای بینایی و کنترل بسیار مناسب و ضروری می سازد. این گونه روشهای پویا منتج به راه حل های قابل انعطاف با پیچیدگی کم و هزینه محاسباتی پایین می شوند. یک عامل رباتیکی برای اینکه قادر باشد که در محیط های بینایی عمل کند باید قادر باشد که فضای ادراکی بینایی خود را به اعمال فیزیکی خود متناظر سازد. این قابلیت ، هماهنگی بینایی – حرکتی ، بینایی مبتنی بر منظور یا بینایی برای عمل نامیده می شود. بر خلاف راه حل های بینایی ماشین که اغلب نمایشهای از قبل تعریف شده و ثابتی در ذهن عامل فرض می کنند ، نمایشهای لازم در بینایی مبتنی بر منظور از تعامل عامل با محیط پیرامونی ایجاد می شوند.

در این رساله ، راه حل هایی برای یادگیری کنترل توجه بینایی بالا به پایین و مبتنی بر وظیفه در محیط های تعاملی و هنگامی که تصاویر پیچیده طبیعی باید پردازش شوند ارائه می دهیم. عامل باید نمایشهای بینایی داخلی اش را همزمان با اعمال فیزیکی خود یاد بگیرد تا قادر باشد یک وظیفه پیچیده را انجام کند. این راه حل ها از ایده های بینایی مبتنی بر وظیفه و توجه بینایی الهام گرفته اند. بطور مشخص در اینجا ما الگوریتم های یادگیری تقویتی ای ارائه می دهیم که قابل اعمال به محیط های بینایی هستند. با الهام از نحوه عملکرد بینایی انسان که پردازش های پیچیده بینایی را بر زیر مجموعه ای از اطلاعات بینایی متمرکز می کند ، هدف در راه حل های ارائه شده محدود کردن استخراج ویژگی بر روی تعداد کمی از نواحی تصویر است. تاکید اصلی بر روی یادگیری توجه مکانی همراه با اعمال حرکتی می باشد.

کلمات کلیدی: بینایی مبتنی بر وظیفه (purposive vision) ، توجه بالا به پایین (top-down attention) ، توجه بینایی (visual attention) ، توجه پایین به بالا (bottom-up attention) ، یادگیری کنترل توجه (learning attention control)، توجه مبتی بر وظیفه (task-driven attention) ، یادگیری تقویتی (reinforcement learning) ، بینایی ماشین (machine vision) ، رباتیک شناختی (cognitive robotics)

پژوهشگاه دانشهای بنیادی

(مرکز تحقیقات فیزیک نظری و ریاضیات)

پژوهشکده علوم شناختی

پایان نامه دکتری

گرایش هوش مصنوعی

یادگیری تعاملی کنترل توجه بینایی مبتنی بر وظیفه

نویسنده : علی برجی

اساتید راهنما :

دکتر بابک نجار اعرابی

دکتر مجید نیلی احمدابادی

بهار ۱۳۸۸

بنام پروردگار دانایی ها

درخت توگر بار دانش بگیرد

بزیر آوری چرخ نیلوفری را