A Developmental and Autonomous Learning Framework for Integrated Active Depth Perception

Tanapol Prucksakorn

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

A Developmental and Autonomous Learning Framework for Integrated Active Depth Perception

Tanapol Prucksakorn

Supervisor: Professor Nak Young Chong

School of Information Science
Japan Advanced Institute of Science and Technology

December 2018

Abstract

Developmental learning is essential for cognitive development. In this research, we examine one of its applications for robots which is active depth perception. Depth perception is one of the most fundamental problems for biological and artificial vision systems. Humans use several different cues to infer the depth layout of a scene or estimate the distance of individual objects. Usually, depth perception in humans is an active process involving different kinds of eye and/or body movements.

During active binocular vision, when an object is fixated with both eyes such that the optical axes of the two eyes intersect at a point on the object's surface, the vergence angle between the two eyes provides an estimate of the object's distance. When the observer moves sideways by a known distance, the eye rotations necessary to keep the object at the centers of gaze, the so-called motion parallax, also provide information about the object's distance. When the observer approaches the object with a known velocity, the changing optic flow pattern created by the movement also provides information about the object's distance. Note that while active depth perception based on vergence eye movements obviously requires at least two eyes, depth perception based on motion parallax or optic flow requires only a single eye. However, humans do not only use one active depth perception for their whole lifetime. They can utilize multiple active depth perceptions when they move. Thus, we consider the full active depth perception which are stimulated when the observer moves in a direction and looking at a specific visual field. All of the three-active depth perception are then evoked as (1) the eye rotation that is necessary to keep the previous visual field to compensate the lateral body movement. (2) the eye rotation required to reduce the disparity between two eyes.

The main goal of the research is to implement a biological inspired active depth perception framework for robots which is developmental and has the ability of self-calibration. A literature review of various studies implementing the vision system indicates that there are several ways to implement the active depth perception. One way is to use the conventional computer techniques to create the depth perception algorithm. Despite their impressive accuracy of the depth perception, most of the frameworks fails to adapt and learn to various environment. So, to solve the problem, some studies proposed the framework with learning algorithms which generally solve the learning issue. However, the studies fail to create a link between action and perception which is important for creating a developmental learning framework.

In this thesis, we describe the works that relate to the research and how we solve the problem with the proposed frameworks such as generating smooth pursuit eye movement when the robot moves in a lateral direction, estimating the distance between the robot and the fixating object with motion parallax, extending the presented visual learning framework to accurately and autonomously represent the various ranges of absolute distance by using the pursuit eye movements from multiple lateral body movements, integrating motion parallax and stereo vision cue within one framework.

Finally, we show that the proposed models, which are implemented in the HOAP3 humanoid robot simulator, can successfully solve the problem that is raised toward achieving the main goal.

Keywords: Active Depth Perception, Cognitive Developmental Robot, Autonomous Learning, Motion Parallax, Self-Calibration, Active Efficient Coding, Integrated Cue, Distance Estimation, Developmental Vision, Eye pursuit, Sensory-motor Coordination

Acknowledgments

I would like to take this opportunity to express my gratitude to those who made a difference in my research life for three years in JAIST. First and foremost, I am deeply indebted to my supervisor, Professor Nak Young Chong, for giving me an opportunity to engage researching at JAIST. He gave very helpful supports and wonderful advices. His guidance helped me in all the time of research.

Besides my advisor, I would like to thank Assistance Professor Sungmoon Jeong for his insightful comments and advices which incented me to widen my research from various perspectives. My sincere thanks also goes to Lee Hosun, and Kshitij Tiwari for their helps, suggestions and encouragements during my research. I would also like to show my gratitude to Professor Jochen Triesch, Vikram Narayan, Alexander Lelais, and Lukas Klimaschutz for the supports form Germany.

Last but no the least, I would like to thank my parents for supporting me to study, research, and writing this thesis here in Japan.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Importance of Research and Its Challenges

With the rise of new developments in robotics and artificial intelligence, they have brought many attentions recently. For examples, Alpha Go [1] won a human world champion in Go, a strategy board game, self-driving car [2, 3] that can learn how to drive by itself without a driver, a study on a very human-like robot that can display emotion [4], and very recent studies of a social humanoid robot, Pepper, that can do various things [5–7].

In the future, we may expect to see many kinds of robot that can learn and interact with us in our daily life like in many movies/novels. However, to reach the vision, a solid foundation of how the robot learns must be established first. Many studies are pursing the vision in many different field such as follows. In [8], they discuss how ubiquitous robotics could be in the far future. It combines the cloud technology which lets numbers of robots share information they learned together. It will open to space of applications such as companion assisting, co-working alongside people, and safety guarding. In [9], they described the state-of-the-art and the future direction of realizing a socialize-able robot which can learn and act alongside with human. Certainly, the development of the surgical robotics could save many lives. In [10] they reviewed the works in the surgical robotics field and highlighting the significant achievements. They described how the research in this field is progressing.

In [11], they described cognitive developmental robotics (CDR), an important key for achieving the vision. In general, there are 3 requirements.

1. Action and perception should be tightly coupled.

2. An agent must be able to learn sensorimotor mapping from experience.

3. An agent must be able to adapt itself to changes.

A robot that satisfied the above requirements should unlock the physical embodiment necessary to be an intelligence system [12,13]. However, most of the studies does not hold all of the requirements (more to discuss on Chapter 2). Therefore, implementing such a system that satisfies all of the conditions while the performance is in an acceptable range is one of the challenge in creating the cognitive developmental robot.

1.1.1 Cognitive Developmental Robotics

CDR gives the keys needed to create such a robot that can learn and perform a variety of complex tasks. It aims to realize and understand human's cognitive functions by synthetic approach since there is little knowledge on the mechanism of the higher order human cognitive functions. To achieve the concept of CDR, physical embodiment is necessary.

In the early stage of the human, experiences gained through interacting with various environments effect how the individual's information structuring such as body and image representation is formed. In the later stage, the individual then may learn by interacting to other agent or being exposed to a new environment. In other words, an individual can learn by obtaining meaningful information through their actions in any form. This concept is what shapes the physical embodiment [14–19].

The studies [20–28] consider the body representation of a robot which associate the visual and tactile sensation that let the robot realize the frame of reference or its own body. The studies share the important key, physical embodiment. They interact to the environment, in this case the robot itself, to gain the information needed to learn its own body representation. [29–38] study the development of joint attention. It simply means two or more agents looking at the same object. The studies share the concept of CDR. The actions generated from the models are gazes, while the information they received is gazes from the other agents/supervisor. There are also other studies that concern CDR such as follows. [39–41] develop model that mimic vocal imitation that baby does. [42,43] proposed models distinct facial expression. [44–46] built lexicon acquisition models.

Recently, there are CDR studies [47–51] that consider gaze control. However, these works do not consider active depth perception under self-induced motion which is one of the abilities that infants use to learn and interact with various things. In this thesis, we further investigate and extend the proposed gaze control models for self-induced motion based active depth perception.

1.2 Motivation and Research Goal

To estimate the distance between a robot and an object, the robot must have depth perception mechanism in order to perceive the depth. There are many ways to estimate the depth such as, stereo vision which is widely used, and there are a lot of researches about stereo vision which give the depth perception ability to robots. However, there is a critical problem.

Most of monocular and binocular depth estimation researches does not only require calibrations before operating, it also requires that the configurations of the system must not be altered. So, if there is any situation or accident that interfere the configurations of the vision system a little bit, the system would begin to fail. Thus, some kind of autonomous and self calibrating mechanism would be needed in those kinds of situation.

In order to make a robot or a vision system that suitable for all environment and robust to interferences, the problems are very crucial and must be solved. A representation of vision system in developed organism, such as human, could be useful to overcome the problems, because humans vision system can adapt to many environment and can recover from interferences. A simple concept of perceiving vision or depth in our human brain is described in Fig. 1.1, the action cycle. The eyes send sensory information to the brain to create vision and depth perception, while the brain learn to control the eyes movement in order to make eye perceive the environment effectively.

The curiosity of creating an autonomous learning active depth perception has been the motivation of this research, such as what are the benefits of implementing the framework? Is it possible to combine the advantages of all active depth perception together by integration? Is it possible to find such a movement that is optimal for active depth perception? If the answers are positive, the research should be able to satisfy the CDR requirements and overcome the critical problem.

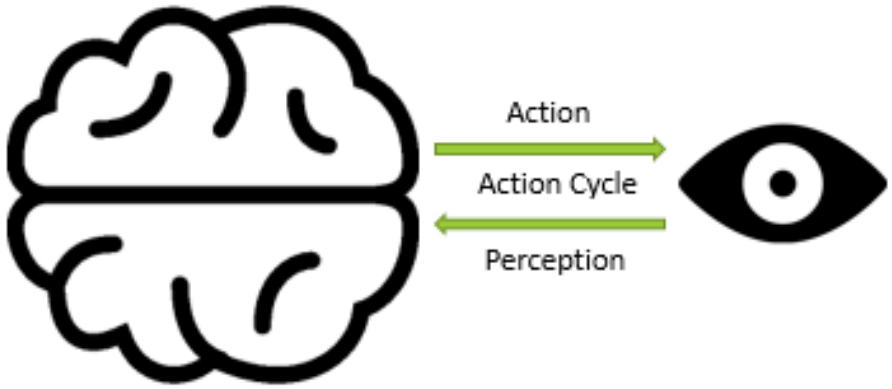


Figure 1.1: Action cycle in most of developed organism

Again, this research aims to satisfy the requirements and overcome the problem. The ultimate goal of the research is to implement a biological inspired active depth perception framework for robots which is developmental and has the ability of self-calibration. The proposed models will contain two important abilities, autonomous learning and self-calibrating. The system will be able to learn how to perform active depth perception. This work will be an another step to create a full representation of biological vision system for artificial vision system.

1.3 Thesis Outline

The organization of this dissertation consists of 9 Chapters. They are organized as follows.

- **Chapter 2** introduces the background and related works of this research. The researches in neural science, robotics, and computer vision field are mentioned and discussed how they are related and motivated to our work.
- **Chapter 3** explains the preliminaries that are required to implement this research.
- **Chapter 4** introduces the framework that the research is based on. It explains how to create the binocular active perception system for a robot.
- **Chapter 5** demonstrates how the framework is extended to understand the motion parallax phenomenon. The robot learns how to generate the smooth pursuit eye movement together with depth estimation.

- **Chapter 6** describes how the framework in Chapter 5 could be improved with the new learning strategy. It compares and analyzes the results with the two different schemes.

- **Chapter 7** questions the predefined lateral movement in the previous chapters that the movement should be learned by the robot. This chapter explains how the robot can learn the optimal lateral body movements.

- **Chapter 8** integrates the two active depth perception cues together which are motion parallax and stereo vision. Dominant eye concept is used to create the unification of the two cues.

- **Chapter 9** Conclude and summarize the research that is done so far. It also shows the contributions of this study and discuss how to further improve the research in the future.

1.4 Summary

The unique points can be summed up as follows: (1) The research focuses on building a mimicked biological vision framework in order to implement the developmental learning vision system for robots, (2) to understand the model underlying in most of the developed organism, (3) unified-learning of action and perception to encourage developmental learning, and (4) the information generated within the framework can be further used for distance and depth perception.

Chapter 2

Related Works

This chapter discusses the related works and where the thesis lies in implementing active depth perception.

The concept of realizing a vision system has been studied extensively in numerous of studies. There are also many applications benefit from the vision system such as mobile robot navigation [52], human-robot interaction [53], and active vision [54].

Remarkably, there are many studies [55–58] that proposed image processing and machine learning techniques to implement depth perception for solving a given task. In [59], they utilize multiple frames captured with a single camera to predict distance. Prediction algorithm is designed and used as a distance estimator under the assumption that the camera motion is known. [60] proposed a biologically plausible visual attention system to selectively localize a salient area. [61–65] proposed image tracking models. [66] used an information theoretic approach to minimize an uncertainty. [67, 68] proposed models to create depth maps from head and eye movement. In [69], they utilized a monocular vision-based obstacle avoidance system by coupling a reinforcement learning together with a linear regression method. [70] combines the triangulation from stereo vision and processed feature from monocular image to yield better depth estimation accuracy.

However, with the aforementioned works, it is quite challenging to create such a system that can develop and adapt itself to the different environments by developing both of perceptual and behavioral abilities at the same time. The main reason is that manual calibrations and prior knowledge are required to finely tune the system during their artificial life.

2.1 Keys to Realize the Biological Inspired Vision System

There are two keys to unlock the biological inspired vision which are developmental learning, and action-perception cycle. By possessing these two keys, it is possible to endow the intelligent behavior to a robot.

2.1.1 Developmental Learning

As discussed in Section 3.1, developed organisms is able to understand the environment around them by learning through their lifetime. The ability to adapt and learn by themselves during their life is usually referred as *developmental learning*. Certainly, a biological inspired system should follow the developmental learning concept. Since this approach and the traditional approach may lead to similar results, it may seems to be unnecessary to develop the developmental system. However, a system that has the developmental learning ability has a larger potential in terms of creating human-like behavior or adapting to various environments, because, for non-developmental systems, robot's configuration and environment are difficult to model and able to change unpredictably [71].

Here are some of the studies that are great examples for having the developmental learning ability. In [72], they proposed a way to implement a developmental learning framework based on work in [73] of eye-head coordination by mimicking the human infants in humanoid robots. They use a constraint-based field-mapping approach for the learning of gaze control. In [74], a convolutional network was used to train vergence eye movements. They use supervised signal to minimize the cost function. [75] proposed a learning model that integrates both static and self motion based visual cues for depth estimation.

2.1.2 Action-perception Cycle

By coupling the action and perception together, the system is able to achieve the physical embodiment, since physical bodies are able to bring the system into meaningful interaction with the physical environment [76]. Visual information improves the robot's behavior, while the resulted actions effectively reinforce the perceptual learning.

In [77–79], they propose a visual servo method to create the link between action and

perception. They use the kinematic connection between the visual information and the camera velocity to realize the action-perception link.

2.2 Active Perception

In the previous section, we showed some of the studies that is related to each key. However, those studies do not have both of the abilities. The visual servo method [77–79] can connect the action and perception together, but it lacks the ability to learn and adapt to different configurations and environments since it need prior knowledge to construct the kinematic link. The gaze control studies [72, 74] achieved the developmental learning ability, however the connection between the action and perception is unclear.

Recently, in [47, 48] they proposed a framework that has the two keys. They use reinforcement learning couple with efficient sensory coding [80–82] to create a vergence eye movement control with a unified cost function, i.e., perception learns to improve behavior and vice versa (joint development). This means that the action and perception are tightly connected resulting an action-perception cycle (Fig. 1.1) which exists in developed organisms.

This has been successfully demonstrated for the case of active binocular vision, where a representation of binocular disparity and the control of vergence eye movements need to be learned. In [83], they also took a similar approach with Gabor filter for binocular disparity coding and Hebbian learning for the eye movement control. In these mentioned studies, the behavior does not simply learn by itself, but it also learns with the help of the perception part, and vice versa. Also, in [50, 51], they showed that extending the framework with the representation of optic flow and pursuit eye movement is possible [50, 51]. Moreover, in [49], they integrated the learning of active stereo vision and active motion vision together. They successfully demonstrated to generate multiple eye movements which are smooth pursuit and vergence eye movements to track an object.

Interestingly, the models are not explicitly trained to perform vergence or pursuit eye movements, but they discover that it is useful to engage in these behaviors, because it improves their coding efficiency. The models encourage the relation between action and perception which are learned by themselves without any supervision.

Furthermore, the scope of this thesis lies in between these works. We propose novel

Table 2.1: Active perception studies versus the key issue. The works below the double horizontal line represent the research in this thesis.

Body Movement	**Key Issue**	**Study**
Stationary	Vergence eye movement control	[47, 48]
	Pursuit eye movement control	[50, 51]
	Vergence and pursuit eye movement control	[49]
Lateral Movement (Motion Parallax)	Motion parallax with Depth Perception	Chapter 5, 6
	Motion parallax with Optimal Movement	Chapter 7
	Integration of stereo vision and motion parallax	Chapter 8

frameworks by extending the previous studies [47–51] with self-induced motion parallax. Also, we propose a new strategy to integrate stereo vision and motion parallax cues together by utilizing dominant eye concept. To list the contribution of each study, we can see Table 2.1.

2.3 Summary

This chapter presented the background of some of the studies that have attempted to implement a vision system. The studies are effective and specialized in their own way. To summarize, the mentioned works are categorized as shown in the Table 2.2. However, with the traditional computer vision approaches, it is not possible to mimic the biological vision system which has the ability to adapt and learn. For the developmental learning approaches it can learn by their own to achieve the biological-like vision system. But, to realize the completed biological vision system, the action-perception cycle is required. Thus, the studies that falls in the highlighted cell is preferred for creating the biological inspired vision system.

Table 2.2: The mentioned studies in this chapter are categorized as whether they are developmental or has the action-perception (AP) cycle.

	Non-Developmental	**Developmental**
Without AP cycle	[55–60, 66, 70] Traditional computer vision	[72, 74, 75]
With AP cycle	[77–79]	[47–51, 83] This Thesis

Chapter 3

Preliminaries

This chapter explains the concept, tools, and algorithms that are used in this research.

3.1 Developmental Learning and Active Depth Perception

For living organisms such as humans and mammals, when they were born they do not instantly understand how to use the information they perceived. They continuously learn and improve their perception while interacting with the environments during their lifetime. This is usually described as developmental learning.

The essence of building the biologically plausible robot is based on the developmental learning of perceptual and behavioral abilities from humans and developed organisms. Recently, there are many studies on computer vision related to human cognitive systems for autonomous robots, inspired by the facts that humans can autonomously develop and recover their perceptual and behavioral abilities to survive in various environments. These abilities are not only useful for extracting visual information for guiding actions, but they are also for perceiving the environments.

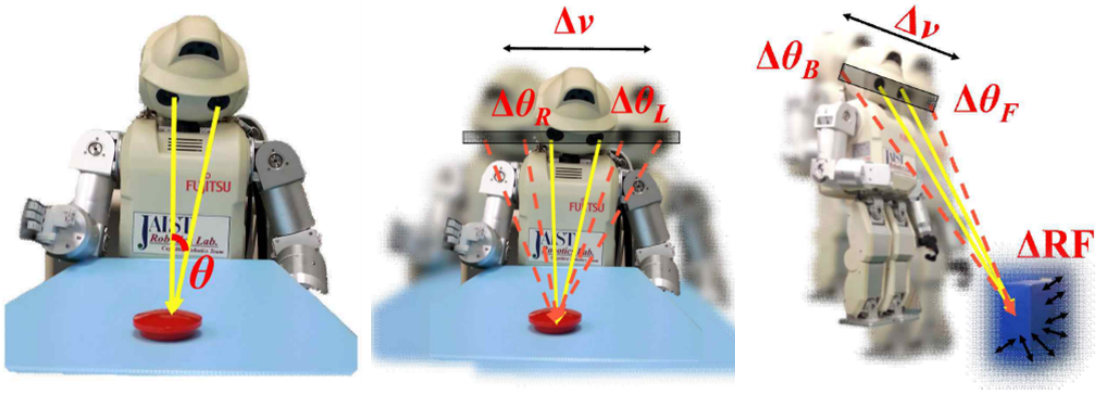
However, the data that are collected by human or animals organs are very noisy messy data. It is not self-explanatory meaningful information [84, 85]. In [86], they discussed that our brain did not programed to know how to use those data, but instead the brain is trained autonomously to learn how to translate those noisy unordered information into useful information.

Synthetic approaches based on explanation and design could be proposed to overcome the shallow knowledge [11], but it is still a very challenging task to implement the developmental system in an autonomous learning manner. In order to realize a developmental robot, a system should equip the two important learning principals which are (1) autonomous development through their artificial life and (2) unified-learning of action and perception. By establishing a tight connection between action and perception, the visual information can be used to improve the robot's behavior [76], while the resulted actions effectively reinforce the perceptual learning.

The same idea also applies to the active depth perception which is a process of producing different kinds of eye and body movement to utilize active visual depth cues. Moreover, it is required that several cognitive developments such as visual representation (sensory coding), eye movement control (action strategy), and depth representation (high-level sensory perception) should be simultaneously performed by integrating each other during their lifetime (life-long learning). However, the underlying ideas of the active depth perception are still unclear.

Depth perception is a visual ability to perceive the world in three dimensions and the distance of an object. Depth perception is the most fundamental artificial vision problem that must be solved. It is an active process that can involve different kinds of movement such as eyes movement, head movement, and body movement. By adapting the biological vision systems with the current artificial vision systems, we can get rid of the dearth of robustness. In neural science, to use information from sensory system, the system should efficiently encode the sensory information by taking advantages of redundancies. So, we may use the nature of the sensory systems in humans to adapt with our artificial vision system. Neurons are the cells that are in our body. They have an ability to propagate signals rapidly over large distances. Sensory neurons fire sequences of action potentials in various temporal patterns to change their activities. To resemble the neurons in our body, sparse coding is used to represent sensory inputs [87].

Active depth perception is depth perception with action of the agent [88], such as movement of the eyes, body, or manipulating the object. There are three approaches of active depth perception (Fig. 3.1). The first one is depth estimation based on the vergence angle between two eyes [89], or stereo vision [90] (Fig. 3.1a). The second one is estimation



(a) Vergence (b) Motion parallax (c) Optic flow

Figure 3.1: Three different depth perception

based on motion parallax [91, 92]. A controlled lateral movement produces a change of the angle under which the object is perceived (Fig. 3.1b). The depth can be estimated by this change. The last one is estimation based on optic flow [93, 94]. The pattern of optic flow or the visual size of the targeted object could be used to estimate the depth (Fig. 3.1c).

3.2 Sensory Coding

Sensory coding is the information processing that is occurring in nervous systems. Signals from each individual neurons are combined and converged to be be processed at the higher levels in the central nervous system to achieve a specific task such as recognizing an object in visual cortex for the visual sensory. To mimic and realize such a system, there are some hypotheses. In this research, we consider the efficient coding hypothesis in order to realize the visual sensory coding for active perception.

The efficient coding hypothesis [80–82] states that sensory systems should encode sensory information in an efficient manner by exploiting redundancies in their inputs. The idea is very promising and lead to numbers of research. For example, it inspired a substantial amount of work on the statistics of natural sensory signals and how they may explain receptive field properties of sensory neurons in visual, auditory, or olfactory systems. This idea was used to extended to associate with active perception that involves the movements of the sense organs, because the statistics of sensory signal will always be

the product of the sensory environment, the characteristics of the sense organs and the agent's behavior. Thus, it is possible to improve the sensory coding by learning to move sense organs in an optimal way.

3.3 Reinforcement Learning

In machine learning, we treat an environment as Markov decision process (MDP). Since a real-world environments are very complicated and involves many variables, reinforcement learning does not aim to fully realize the whole environment, but to simulate them with out prior knowledge about the environment model (unsupervised learning). This makes the reinforcement learning suit to this research. Through out this thesis, we use a reinforcement algorithm to represent the learning of behavior of the robot.

Reinforcement learning define a policy which maps the state of the actor in its environment to a specific action. The main concept is that an agent (the robot) do something, then it receives a reward with respect to the selected action (training) such as in Fig. 3.2.

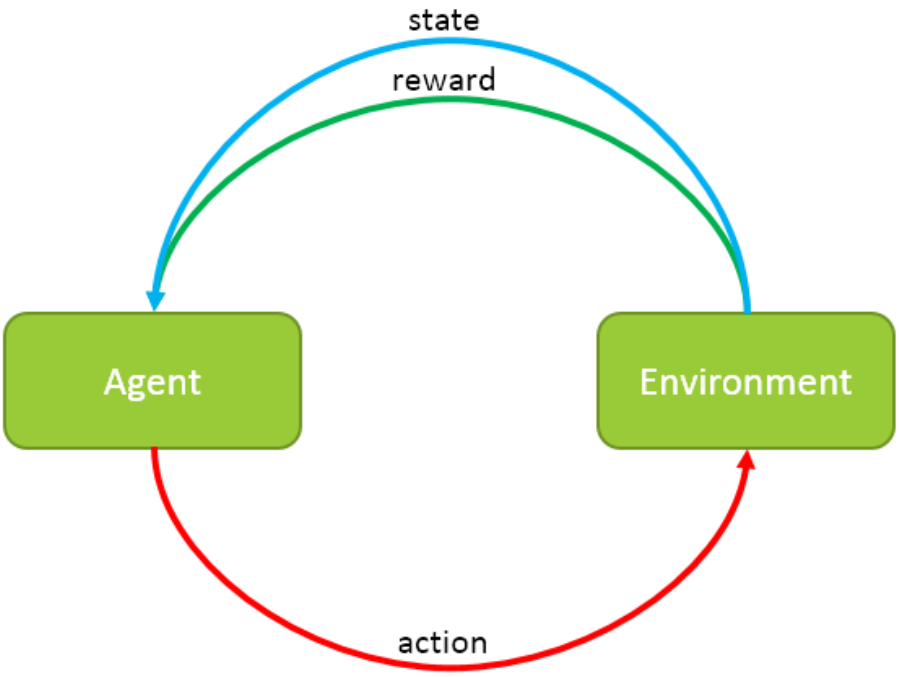


Figure 3.2: Basic diagram of reinforcement learning

The foundation of every reinforcement learning model is that it has a set of environ-

ment states S, a set of actions A, and policy. The flow of the steps is as follows:

1. Observe state, s_t
2. Decide on an action, a_t
3. Perform action
4. Observe new state, s_{t+1}
5. Observe reward, r_{t+1}
6. Learn from experience
7. Repeat step 1

The agent aims to find a suitable policy that maximizes the observed rewards over its lifetime. It considers two important functions. Value function evaluates the best rewards that the agent could get in its lifetime based on its action in the past.

$$V^{\pi}(s_t) = R(s_t, \pi(s), s_{t+1}) + V^{\pi}(s_{t+1}) \tag{3.1}$$

Where, $R(s_t, \pi(s_t), s_{t+1})$ is the reward that the agent would get, if the agent perform action at state s_t with respect to the policy π to the state s_{t+1}.

The another function is a state-action value function $Q(s_t, a_t)$. It is different from the previous value function $V^{\pi}(s_t)$. It shows the best reward that the agent could get if take the action a_t from state s_t.

$$Q(s_t, a_t) = R(s_t, a_t, s_{t+1}) + \max_{a'} Q(s_{t+1}, a') \tag{3.2}$$

In this research, we use the Natural Actor-Critic Reinforcement Learning algorihtm [95], a modified actor-critic reinforcement learning. The next section will explain the basic of the actor-critic reinforcement learning and then the last section will explain the algorithm we use.

3.3.1 Actor-Critic

Actor-Critic is a reinforcement learning that consider two subconscious mind which are actor and critic. The actor generate an action based on the critic, by mapping states to

actions based on probabilistic. The critic criticize the action selected by the actor, by mapping states to expected cumulative future reward. In other word, the critic consider a prediction problem, while the actor focus on the control of the action. Both actor and critic shares the same error which is temporal difference (TD) as shown in Fig. 3.3. The error is used to estimate the average reward for a state-action pair. TD error, δ_t, is defined by

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{3.3}$$

Critic: an action a_t is strengthened based on the TD error. TD error measures the selected action. Positive TD error means that the selected action a_t has a better reward, so the action a_t should be encouraged in the future. While, a negative TD error discourage the action a_t.

Actor: actor use the information from the critic to update the policy parameter of the actor, $\theta(s_t, a_t)$ as follows:

$$\theta(s_t, a_t) = \theta(s_t, a_t) + \beta\delta_t \tag{3.4}$$

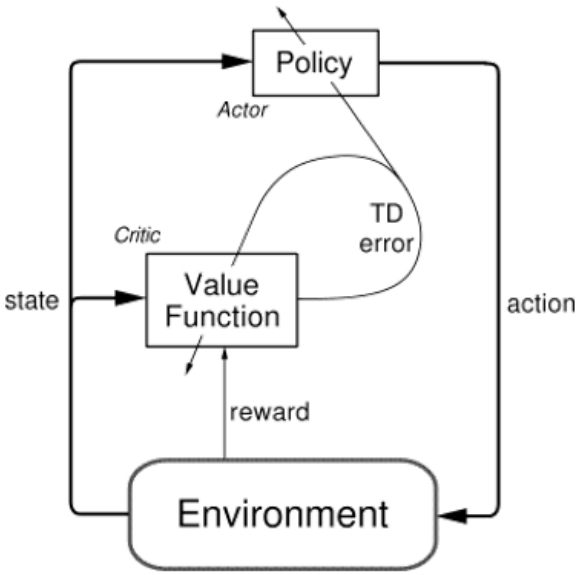


Figure 3.3: Actor-Critic model

3.3.2 Natural Actor Critic

Natural actor critic proposed in [95] is considered as a reinforced actor critic reinforcement learning algorithm. They provide 4 variations of the algorithm. In this research, we choose the algorithm number 3, since it is also recommended by the creator themselves.

Natural actor critic uses two linear neural networks to implement the actor and the critic (Fig. 3.4).

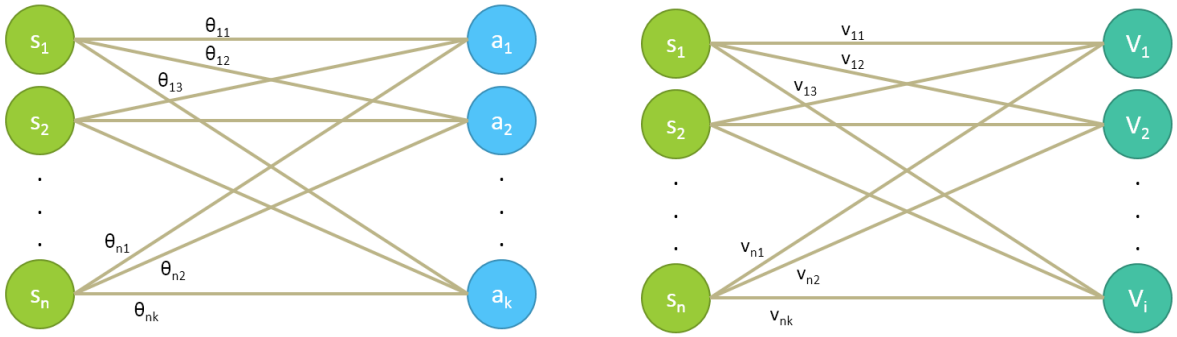


Figure 3.4: Two neural network implementing actor and critic

The selected algorithm is explained in Table 3.1 below.

- t is the iteration number.
- $\hat{J}$ is average reward.
- f_{s_t} is a feature vector for state s_t.
- v is neural network weights for feature vector f_{s_t}.
- w is neural network weights for policy parameter vector θ.
- θ is policy parameter vector.
- α, β, ξ are step sizes for updating weight vector w, θ, and average reward $\hat{J}$ respectively.
- $\phi_{s_t a_t}$ is a feature vector for state-action pair.

for softmax activation policy, Gibbs distribution, which we use in this research

$$\pi(s_t, a_t) = \frac{e^{\theta^\intercal \phi_{s_t a_t}}}{\sum_{a' \in A} e^{\theta^\intercal \phi_{s_t a'}}} \tag{3.5}$$

Table 3.1: Natural Actor Critic Algorithm 3 in [95]

1:	**Input**:	
	• Randomized parameterized policy π	
	• Value function feature vector f_s	
2:	**Initialization**:	
	• Policy parameters $\theta = \theta_0$	
	• Value function weight vector $v = v_0$	
	• Step sizes $\alpha = \alpha_0, \beta = \beta_0, \xi = c\alpha_0$	
	• Initial state s_0	
3:	**for** $t = 0, 1, 2, ...$ **do**	
4:	**Execution:**	
	• Draw action $a_t \sim \pi(s_t, a_t)$	
5:	**Average Reward Update:**	$\hat{J}_{t+1} = (1 - \xi_t)\hat{J}_t + \xi_t r_{t+1}$
6:	**TD Error:**	$\delta_t = r_{t+1} - \hat{J}_{t+1} + v^\intercal f_{s_{t+1}} - v_t^\intercal f_{s_t}$
7:	**Critic Update:**	$v_{t+1} = v_t + \alpha_t \delta_t f_{s_t}$
		$w_{t+1} = [I - \alpha_t \psi_{s_t a_t} \psi_{s_t a_t}^\intercal] w_t + \alpha_t \delta_t \psi_{s_t a_t}$
8:	**Actor Update:**	$\theta_{t+1} = \theta_t + \beta_t w_{t+1}$
9:	**endfor**	
10:	**return** Policy and value function parameters θ, v	

$$\psi_{s_t a_t} = \phi_{s_t a_t} - \sum_{a'_t \in A} \pi(s_t, a'_t) \phi_{s_t a'_t} \tag{3.6}$$

3.4 Neural Network

Neural network is an artificial systems that is inspired by the biological neural networks that can mostly be found in the developed organisms. The system learns to achieve the given tasks by accounting the given supervised examples. It can be used to predict or estimate a specific value if it is given enough of the examples.

Throughout the thesis, we consider only a basic neural network which contains only 3 layers of the artificial neurons for realizing the depth perception module. We chose

the neural network because it suits to our goal which is creating the biological inspired framework for robots.

Chapter 4

Realizing of Active Perception

This chapter explains the fundamental of each module in the framework proposed in [47, 48] which are the foundation of our research.

4.1 Philosophy of This Work

Before we start to explain the work in detail, we would like to emphasize the philosophy of the work. The work aims to generate the vergence eye movement that is require to minimize the disparity of the perceived images from the two cameras. This task can be done with the conventional computer vision techniques as discussed in the introduction chapter. However, the studies either lack of the link between action and perception or are non-developmental system. So, it is difficult to mimic the developing organisms visual systems which has the processes tightly coupled.

4.2 Vergence Eye Movement

When we look or focus at an object, the line of sights of the two eyes cross. The movement that is required to achieve that is called vergence eye movement. It is a simple function that control both eyes to point their fovea on a visual stimulus. Both eyes rotates in opposite direction to maintain the same binocular vision (Fig. 4.1). In case of the robot, two cameras are rotated around the vertical axis (pan) so that the observed images are similar.

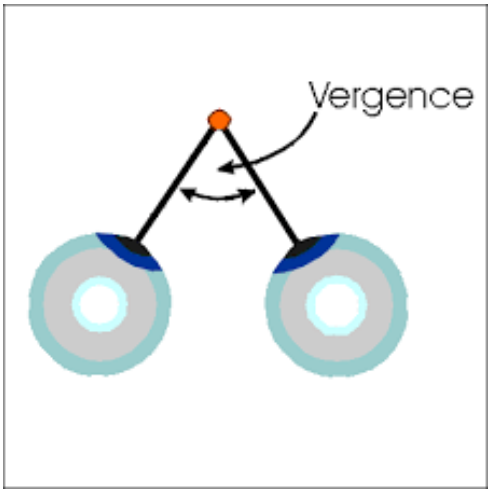


Figure 4.1: Vergence eye movement

4.3 Model Architecture

The work utilizes the active efficient coding theory together with a reinforcement learner to create the binocular active perception framework. It focuses to generate appropriate eye movements to fixate a visual stimulus for both eyes (vergence eye movement). Their framework is shown in Fig. 4.2. Two images are taken from two cameras and then input to the sensory coding model. Sensory coding model represent as the perception module of the robot. It encodes the input images into a sparsely encoded image vector which is then sent to the behavior module, the reinforcement leaner. Reinforcement learner learns to select a vergence eye movement based on the unified cost function which is the reconstruction error from the sensory encoder. This cost function measures how efficient the images could be encoded. Low reconstruction error means that the two input images are similar, thus the two images has low disparity, i.e. the two cameras are pointing at the similar point. The following sub-sections will explain the details of each module.

4.3.1 Sensory Coding Model

As discussed earlier in Chapter 1, a sensory system should encode sensory information efficiently by exploiting redundancies in their inputs. A sparse coding technique is used to implement the sensory coding model under the active efficient coding theory. It encodes the input images from the two cameras into a one dimensional sparse vector. It also computes the loss or the reconstruction error of the encoded vector. Each part of the

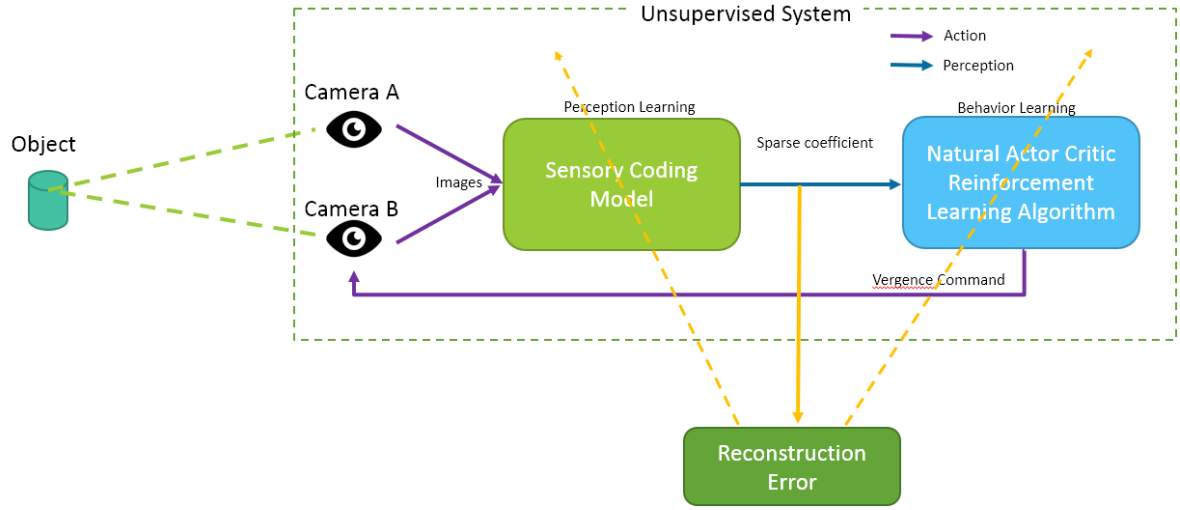


Figure 4.2: Zhao et al.'s framework

sensory coding model is shown in Fig. 4.3

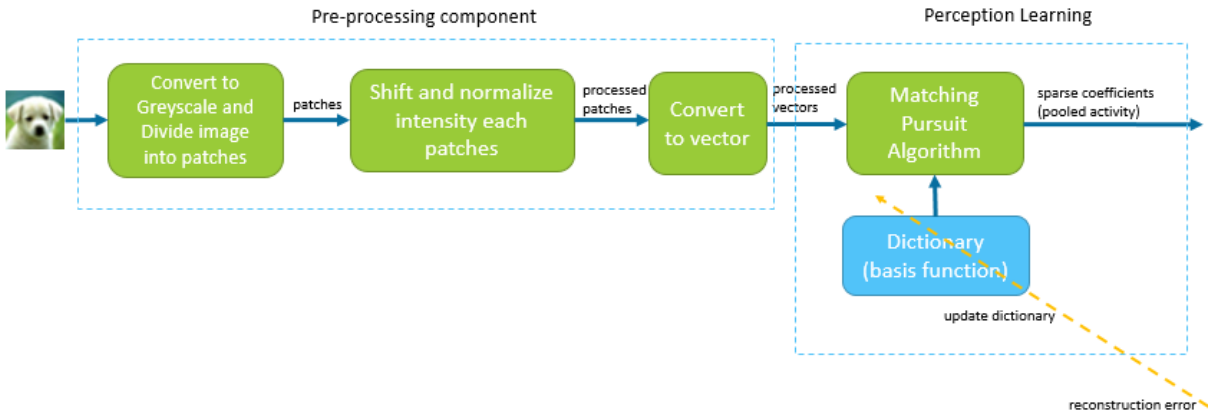


Figure 4.3: Inside of sensory coding model

The inputs are the images taken from the two cameras. Disparities are measured by the horizontal shift of image of the object from left to right. Different depth makes different disparity. To understand the framework, we recreate the framework in a simulation. 6 images are used to test the framework (Fig. 4.4).

Throughout the simulation, the images are selected randomly every 10 iterations. To virtually simulate the image taken from cameras, the images are cropped with the windows size of 128 by 128 pixels in the center of the images. There are two crop windows to represent each camera. One window is fixed at the center, while the another window

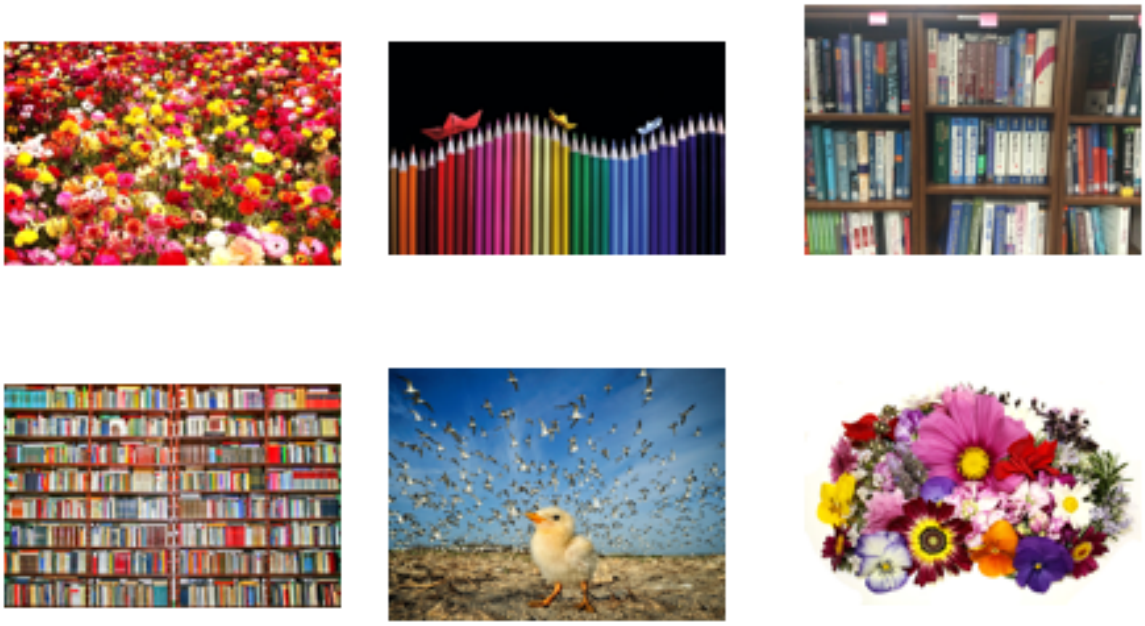


Figure 4.4: Images that are used in binocular vision framework simulation

can be shifted horizontally to represent the vergence eye movement, i.e. the stereo pairs are artificially generated by shifting one of the input images horizontally. The goal of the framework is to generate the vergence eye movement that yields close to zero retinal disparity.

The two input images are converted to gray scale. Then, we extract images into a multiple of 8 by 8 pixel patches. The patches are extracted by shifting 1 pixel horizontally and vertically. The patches are then sub-sampled by a factor of 8 by using Gaussian pyramid algorithm. Finally the patches are converted to a vector and normalized to have zero mean and unit norm $x_i(t)$, where i is the index of the patches. The processed image vector from left and right eye are concatenated into a single vector $x(t)$. The vector has $P = 128$ elements.

For the encoding part, the sensory encoder select a linear combination of basis functions drawn from an over-complete dictionary $\phi(t) = \{\phi_n(t)\}_{n=1}^{N}$ to represent the sparse coding [87]. In our setup, we prepare a visual dictionary that contains $N = 288$ randomly generated normalized basis functions. Matching pursuit algorithm is used to estimate and

find the appropriate linear combination as follows:

$$x_i(t) \approx \sum_{n=1}^{N} a_{i,n}(t)\phi_n(t) \tag{4.1}$$

We limit the number of non-zero scalar coefficients $a_{i,n}(t)$ used in the matching pursuit algorithm to 10 elements to enforce the sparsity of the encoded image (efficient coding). The coefficients generated by the algorithm are the final product of this module which will be used later in the reinforcement learner as pooled activity. Pooled activity simply represent the activeness of each neuron (the coefficients). It consider the sum of all coefficient from every patches as follows:

$$f_n(t) = \sum_{i=1}^{P} a_{i,n}(t)^2 \tag{4.2}$$

To measure the redundancy in the input images, we use the reconstruction error which compares between the encoded images and the input images.

$$e(t) = \frac{1}{P} \sum_{i=1}^{P} \frac{\|x_i(t) - \sum_{n=1}^{N} a_{i,n}(t)\phi_n(t)\|^2}{\|x_i(t)^2\|} \tag{4.3}$$

This error can be used to improve the visual dictionary by using the gradient descent updating technique. Importantly, this error will also be used in the reinforcement learner, thus the unified cost function.

4.3.2 Multi-Scale Framework

Binocular cells tuned to different disparity ranges in visual cortex areas. These cells adjust and adapt the controlling mechanism to generate fast or slow vergence response depending to the range of disparity [96]. In [48], they show that the framework proposed in [47] has the input disparities limitation which means that the framework could not generate an appropriate eye movement if the disparity is too high. So, [48] propose a strategy to overcome the problem. They use multi-scale input images to represent two areas, foveal and parafoveal area. The model use two scales of images which are fine scale and coarse scale as shown in Fig. 4.5. Fine scale images represent a foveal region in our eyes, as we can get more detail from the center of vision. The coarse scale represents a parafoveal area.

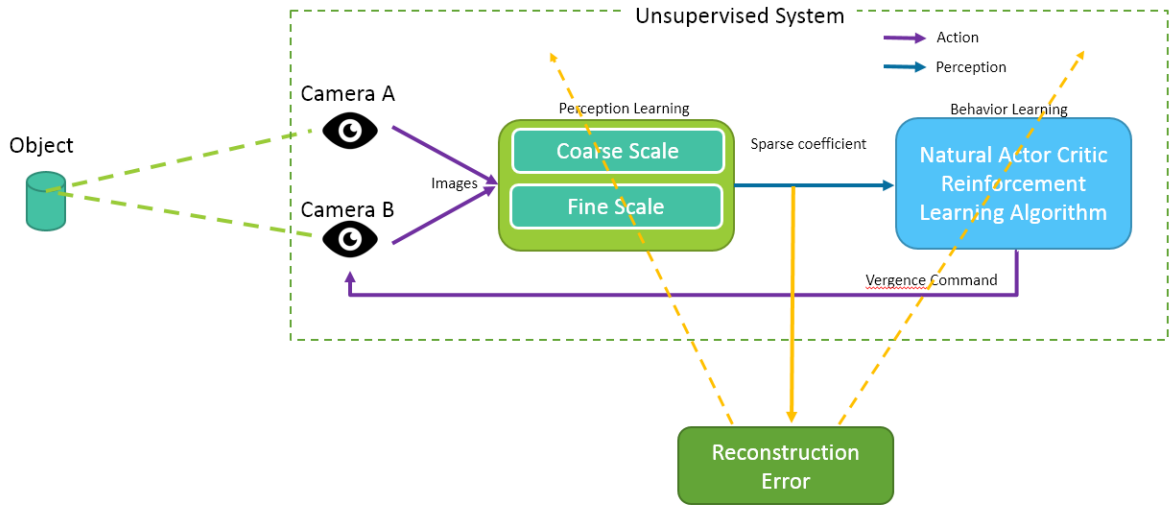


Figure 4.5: Multi-scale binocular vision model

The process is still similar to the above, but with multi-scale we add two 80x80 pixels crop windows to represent the fine scale. The sub-sample factor for this scale is 2. The patches are extracted by shifting 4 pixels horizontally and vertically. An additional dictionary is used to represent the fine scale's visual dictionary.

4.3.3 Reinforcement Learning

Uni-scale Framework

As mentioned in Chapter 3, we use the natural actor critic reinforcement learning algorithm. The state is represented by the pooled activity, while the reward is a function of the reconstruction error (unified cost function). The pooled activities are used as state as follows:

$$f_{s_t} = f(t) = \begin{bmatrix} f_1(t) \\ f_2(t) \\ \vdots \\ f_P(t) \end{bmatrix} \tag{4.4}$$

Actions that are generated from this reinforcement learner are the vergence eye movements. Negative of the reconstruction error is used as the reward to train the reinforcement learner.

$$r_t = -e(t) \tag{4.5}$$

The reinforcement learner learns to select actions that maximize the discounted cumulative future reward, i.e. minimizing the reconstruction error.

Actions are defined as $A = \{-2, -1, 0, 1, 2\}$. The elements in A represents the number of pixels to be horizontally shifted (virtual vergence eye movement). We use Gibbs distribution for choosing an action. In this simulation, the step sizes are set as follows:

- $\alpha = 0.1$
- $\beta = 0.01$
- $\xi = 0.01$

The neural network weights v, w, and policy parameter θ are initially randomized.

Multi-Scale Framework

Similar to the above, we use the pooled activities to represent the current state. The difference is we concatenate the two pooled activity together.

$$f_{s_t} = f(t) = \begin{bmatrix} f_1^C(t) \\ f_2^C(t) \\ \vdots \\ f_P^C(t) \\ f_1^F(t) \\ f_2^F(t) \\ \vdots \\ f_P^F(t) \end{bmatrix} \tag{4.6}$$

The reward is also modified to consider the sum of the error from both scales.

$$r_t = -(e^C(t) + e^F(t)) \tag{4.7}$$

Superscript F means fine scale, while superscript C means coarse scale. We use the set of actions, step sizes, and softmax operation as the same in the uni-scale framework simulation setup.

4.4 Simulation & Results

We use mean absolute error (MAE) to measure the eye movement performance. It tracks the vergence error in the iteration before the image is changed which is every 9^{th} iteration. MAE is defined as follows:

$$MAE(t) = \frac{1}{100}\sum_{k=0}^{99} |\alpha(t+9+10k) - \alpha^*(t+9+10k)| \tag{4.8}$$

where α^* is the target vergence at the current iteration. Fig. 4.6 shows the eye movement MAE from the simulation.

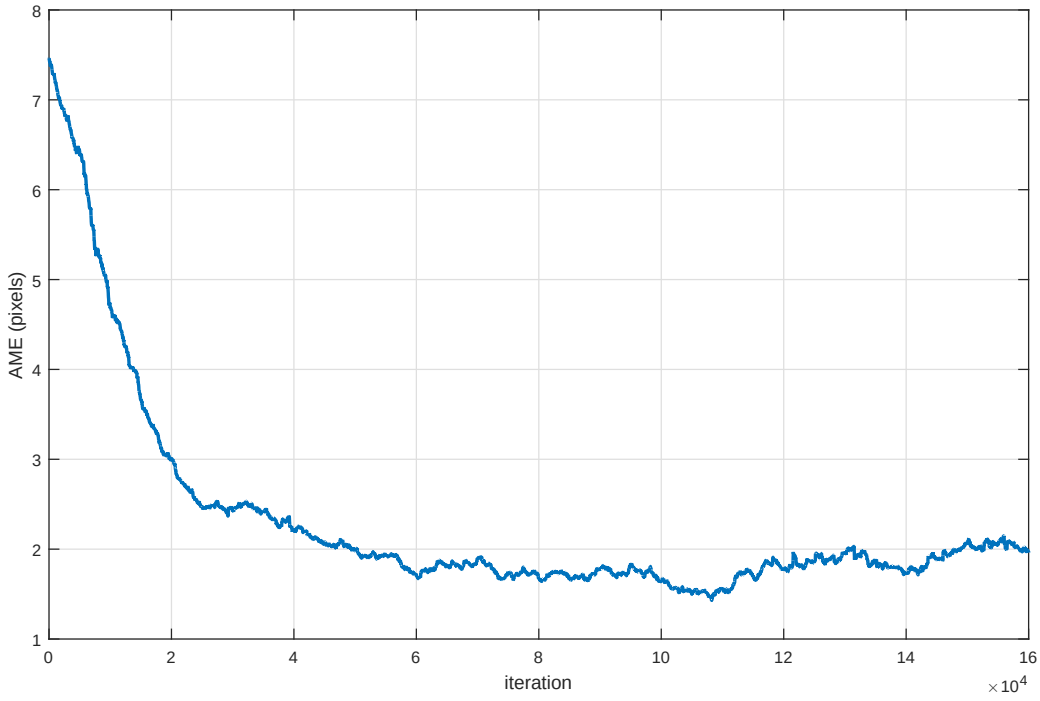


Figure 4.6: MAE of the simulation

The error reduces over time and stays around 2 pixels. This means that with the current specification, the framework has learn its best from the available input images. After the framework is trained to generate the overlapped images (Fig. 4.7), we perform another test to evaluate the framework at different disparities. Fig. 4.8 shows the actual vergence eye movement versus the desired vergence, while Fig. 4.9 shows the vergence error.

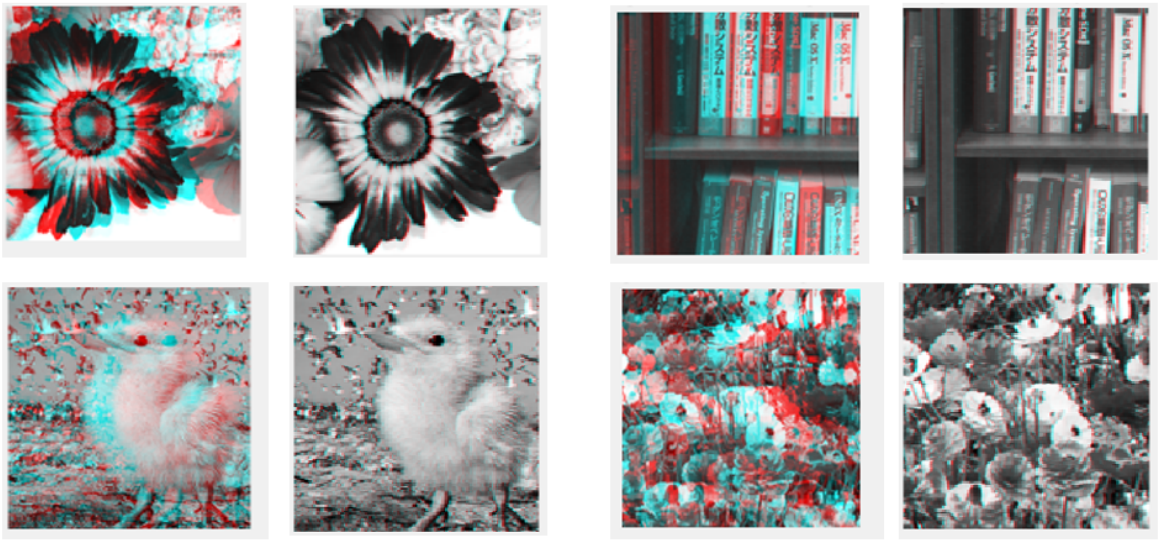


Figure 4.7: Example of some of results of the simulation

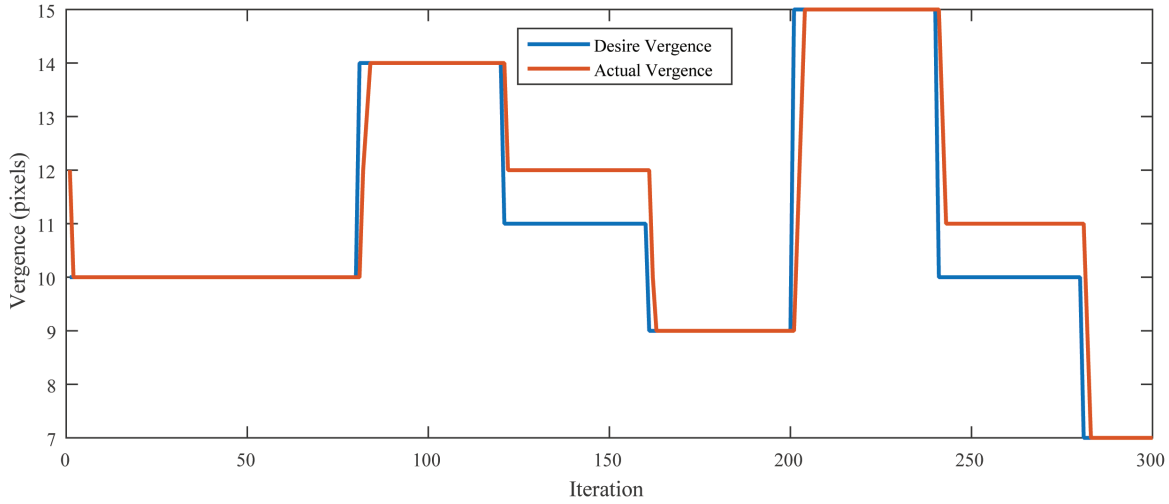


Figure 4.8: Vergence tracking after training is finished

We can see that the framework can properly control the vergence eye movement. It can handle the quick changes in disparity. It can maintain the disparity after reaching zero retinal disparity. The maximum error is around 1 pixels.

4.5 Summary

In this chapter, a novel framework proposed in [47] and multi-scale extension of the framework [48] have been explained and discussed. We showed that the system can

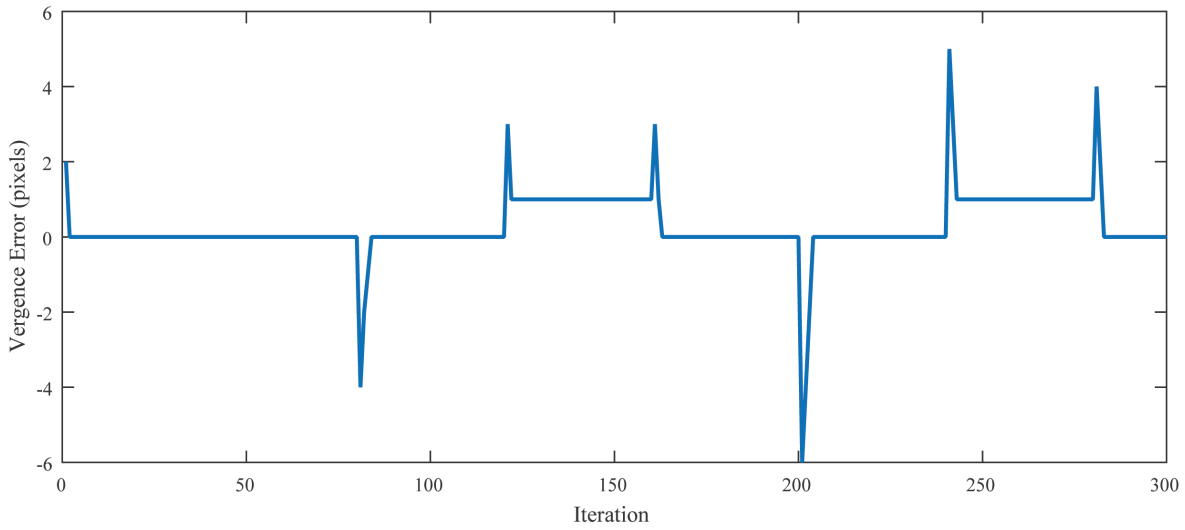


Figure 4.9: Vergence error

autonomously learn how to control left and right camera to generate vergence eye movement.

Chapter 5

Schemes of Motion Parallax Based

After we have studied and understand the previous framework in the chapter 4, this chapter explains the extended framework with motion parallax.

5.1 Philosophy of This Work

Estimating depth by using vision system has been continuously researched for a long time. There are a lot of works that can estimate depth by using binocular disparity. However, there is little work on depth estimation by using monocular depth cue. Some of them require specific condition such as environment, some requires calibration. So, if there are some changes or interferences in environment or configuration of vision system, the solution seems to fail later. In order to overcome this problem, we extend the framework in the chapter 4.

The proposed model will have two important abilities, autonomous learning and self-calibrating. The system will be able to learn how to generate an appropriate eye movement during lateral movement for fixating an object. Finally, this chapter will show that extending the stereo active depth perception to another kind of active depth perception is possible.

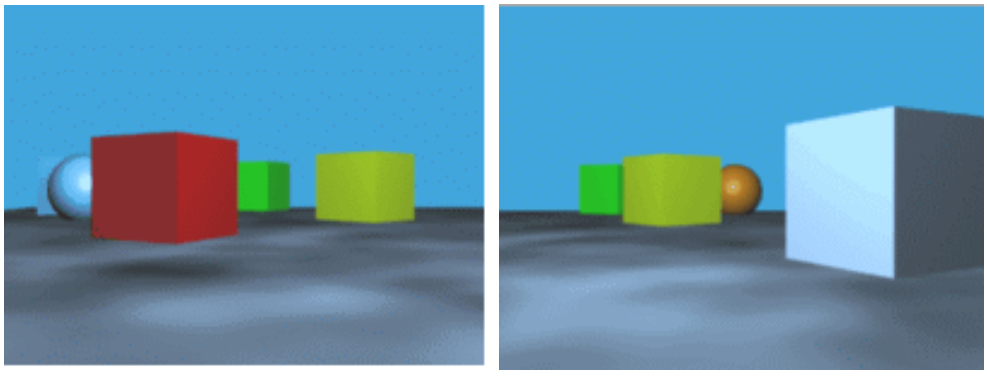
5.2 Motion Parallax

Parallax is derived from "parallaxis", a Greek word which means alteration. It is used in many application such as in astronomy for measuring distances to the closer stars.

Combining with motion we get a phenomenon which happens when we move laterally. It let us to perceive the apparent position of an object from two different viewpoints.

It is an important effect that can be observed in daily life. It gives us useful information that helps to learn and understand the surrounding environments. When we moves in a lateral direction, we can observe various ranges of motion parallax effect occur by maintaining the visual fixation on an object. We perceive close object to move faster than the object that is farther as shown in Fig. 5.1. For an example, at start we can see red box and yellow box (Fig. 5.1a), but after we moved laterally we can see only the yellow box (Fig. 5.1b). We can conclude that yellow box is farther than the red box.

Usually, motion parallax effect provides two different kinds of depth perception which are the distance from the observer to the fixating object (egocentric distance), and the distance from the fixating object to another object (allocentric distance). Usually, allocentric distance is extracted from the motion parallax phenomenon such as in [97] they discuss how it is possible to generalize the relationship between the eye movements and the allocentric distance. However, that is not only the strong point of utilizing the motion parallax effect. In [98], they show that it is possible for humans to extract the egocentric distance. Also, the retinal motion induced by the motion parallax effect can be utilized to observe the apparent depth (egocentric distance) appears on the sagittal plane [99]. For simplicity, in this thesis, depth and distance mean the egocentric distance.



(a) Start position (b) End position

Figure 5.1: Images created by lateral movement from left to right

5.3 Smooth Pursuit Eye Movement

To maintain fixation on an object, a smooth pursuit eye movement is required. It is simply an eye movement that keeps track of a visual stimulus. As shown in Fig. 5.2, the eye must rotate in the opposite direction of the movement to maintain the fixation. If the subject move to the left, the eye must rotate to the right, and vice versa.

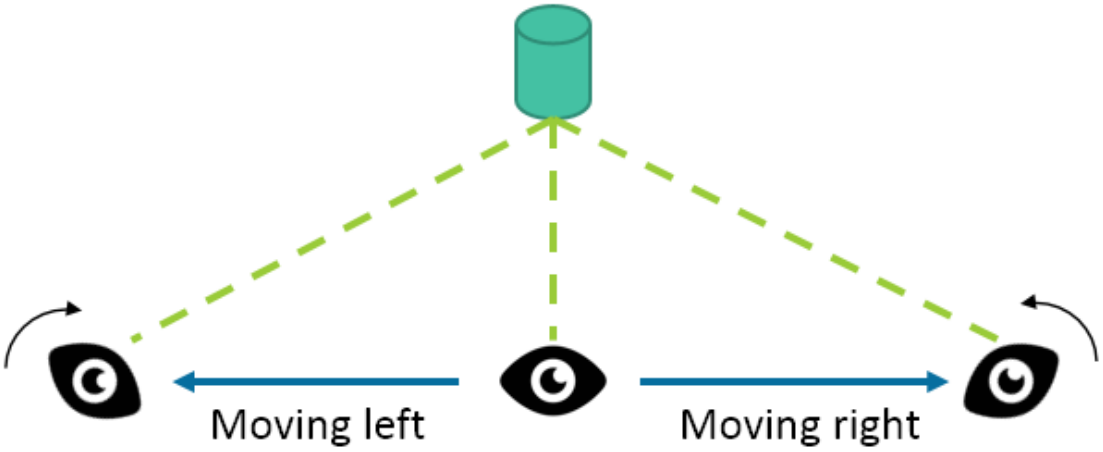


Figure 5.2: Depth perception from motion parallax

5.4 Model Architecture

We consider different image input and camera control to extend the framework as shown in Fig. 5.3. Also, we only use one camera for the image input, since to achieve motion parallax one eye is sufficient. Two different viewpoint is achieve by moving camera laterally. The images from the two viewpoints are used as the input images. The output of the reinforcement learner is the smooth pursuit eye movement. The goal of the framework is to generate the smooth pursuit eye movement to fixate the object at the center of the gaze after moving laterally. Then the movement information is used to estimate depth by using a two layer neural network. The neural network is supervised. It use the ground truth depth information.

5.5 Experiments & Results

We verify the framework with MATLAB and a robot simulator called V-REP. The main framework and algorithm are implemented in the MATLAB, while V-REP provides the environment simulation, Fig. 5.4. The simulation environment composes of a HOAP3 robot, an interchangeable texture box, and a background.

5.5.1 Simulation

Lateral movement of the robot is simply pick-and-place. The possible distance between the box and the robot are 1 meter to 2 meters. The robot moves laterally from left to right by 50 centimeters for 5 steps, thus we get 5 images for one lateral movement from left to right, Fig. 5.5. Two successive images are used as the input images. Similar to the previous chapter, we use the image shifting to simulate the eye movement.

After processing the first two successive images for 15 iteration, the sensory coding model selects the next pair of the successive images. After reaching the final pair of the successive images, the texture box is picked an d placed farther by 10 centimeters. The process is repeated until the depth reaches 2 meters, then the depth is reset to 1 meter. Every 14 iterations, we record the number of shifting pixel q (eye movements) together with depth d at that point of the time in a depth data matrix D.

$$D = \begin{bmatrix} q_1 & q_2 & q_3 & \cdots \\ d_1 & d_2 & d_3 & \cdots \end{bmatrix} \tag{5.1}$$

After the framework can properly generate the smooth pursuit eye movement (by overlapping the two successive images), we train the depth estimation part. Neural network is used to estimate the depth by learning from the depth data D. We use a two layer feed-forward neural network with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer (Fig. 5.6). The hidden layer has 10 neurons. The training algorithm is Levenberg-Marquardt method. In the first row of the depth data matrix D, we use it for the input of neural network. We use the second row of the matrix to be the target. 70-percent of the data is reserved for training. 15-percent is for validating. And another 15-percent is for testing.

Simulation Results

Figure 5.7 shows that the robot is able to generate the smooth pursuit eye movement to fixate the object. The MAE is shown in Fig. 5.8. The error is around 1 and 2 pixels. Even though the result is not perfect, but it is near zero and it is still useful for estimating the depth information.

For the result of the depth estimation, we can see Fig. 5.9. It shows the error histogram of the depth estimation error, which is the comparison between real depth and the estimated depth. Each bin contains instances that have error in that range. We can see that most of the error is closed to zero.

After that, we test the depth estimation by varying the depth between 1 and 2 meters in the resolution of 10 centimeters by using the same image texture in the training. Table 5.1 shows the errors. We also perform another test, but we use different image texture than the one we used in the training. The result is shown in Table 5.2.

Table 5.1: HOAP3 simulation result (training depths)

Input Depth (meter)	Output Depth (meter)	Error (centimeter)
1.00	1.02	2
1.10	1.10	0
1.20	1.20	0
1.30	1.27	3
1.40	1.47	7
1.50	1.47	3
1.60	1.60	0
1.70	1.81	11
1.80	1.86	6
1.90	1.91	1
2.00	1.99	1

We can see that the framework is able to estimate the depth of the texture box with small errors. Although at some depths, the estimated depths are the same. The reason behind this is that there is a little offset error of the eye movements. And the difference of disparity between each depth is quite small or similar. So, for depths that are close

Table 5.2: HOAP3 simulation result (random depths)

Input Depth (meter)	Output Depth (meter)	Error (centimeter)
1.25	1.29	4
1.53	1.60	7
1.77	1.86	9
1.92	1.90	2

together, it is possible that the results are the same. This problem could be eased by increasing the image resolution and the patch size, however a higher computational power is required.

5.5.2 Real Hardware Experiment

After we verified the framework in the simulation, we test the framework in the real world.

Hardware Setup

The setup that we use in this experiment is shown in Fig. 5.10. We use MATLAB to run the framework. A micro-controller, Arduino, is used for receiving command from MATLAB and controlling an XY-table. A camera is attached to the XY-table.

However, motion parallax requires only lateral movement, so we use only one axis of the XY-table. The flow of the system is the same as in the simulation, except that the camera and camera controlling part are in the real world.

In this experiment, we have the XY-table and object on a floor. As shown in Fig. 5.11, the camera (blue eye symbol) can move laterally to generate the motion parallax images. The depth between the camera and the object (black cube with red stripe) will be varied by hand manually. In this case, the camera will move laterally for 12 centimeters. Each step move for 3 centimeters, thus we have 4 images per lateral movement. The view from the camera is shown in Fig. 5.12.

In order to make training easier, we gather all of the data required to train before run the training. We capture all images generated from lateral movement in various depth, from 40 centimeters to 1 meter (each step increased by 10 centimeters). Then we use the set of images that we have gathered to train the framework.

Experimental Results

Fig. 5.13 shows an example of tracking of object from two successive frames. Fig. 5.14 shows MAE of pixel shifting.

Fig. 5.15 shows error histogram of the trained neural network. We test the framework in the same way as in HOAP3 simulation. The results are shown in Table 5.3, and Table 5.4.

Table 5.3: Experimental result (training depths)

Input Depth (centimeter)	Output Depth (centimeter)	Error (centimeter)
40	35.34	4.66
50	49.40	0.60
60	54.33	5.67
70	72.18	2.18
80	83.45	3.45
90	88.76	1.24
100	91.98	8.02

Table 5.4: Experimental result (random depths)

Input Depth (centimeter)	Output Depth (centimeter)	Error (centimeter)
45	47.90	2.90
65	60.35	4.65
85	88.76	3.76

The experimental results are similar to the simulation results in the HOAP3 simulation. As discussed in the simulation section 5.5.1, we can increase the resolution of the input images and patch size to increase perceivable depth resolution. But, it comes with costs of computation time.

5.5.3 Robustness Test

The simulation and experiment show that our system is able to learn to generate eye movements to stabilize the object in the image center. To demonstrate that the system

has an ability of developmental learning, we apply a perturbation to the system. We rotate the camera clockwise by 20 degrees and keep training the system. The rotation of the camera for the real world experiment is simulated by rotating the input images. As shown in Fig. 5.16, a noticeable increase in error occurs when the perturbation is applied. The figure also shows that the system can recover from the perturbation, i.e. the system is able to learn to adapt to the changes in configuration.

5.6 Summary

This research extends the recent works on self-calibration of active motion vision. We applied it to create a model that learns to keep fixating an object when the camera is moving laterally. We showed that we can utilize the eye movements for estimating the depth of an object by using a neural network. The difference from their works is that we consider self-induced motion parallax, which helps the system to extract depth information.

According to the simulation results, the proposed framework can successfully estimate depth and generate eye movements to keep the object at the center of gaze. Both action and perception learning are trained by the same reconstruction error function. The framework can simultaneously learn to choose actions and create visual representations to understand the motion parallax effect. Moreover, the proposed model can be applied with any single camera system, because it does not depend on details of the hardware and its configuration.

The extended framework can also be described as the low-level visual cue in the primary visual cortex (V1) [100], as it only focuses on maximizing the sensory encoding efficiency (sparse coding) of the available visual stimulus. Since allocentric depth requires a higher understanding of the concept of the object such as border ownership which is represented by some of the V2 and V4 neurons in the visual cortex [101], this research focuses on observing the egocentric distance.

In conclusion, we have proposed a method to extend the active depth perception of original framework proposed in [47]. In addition, we proposed a method to use the information from motion parallax to estimate the depth between camera and the object.

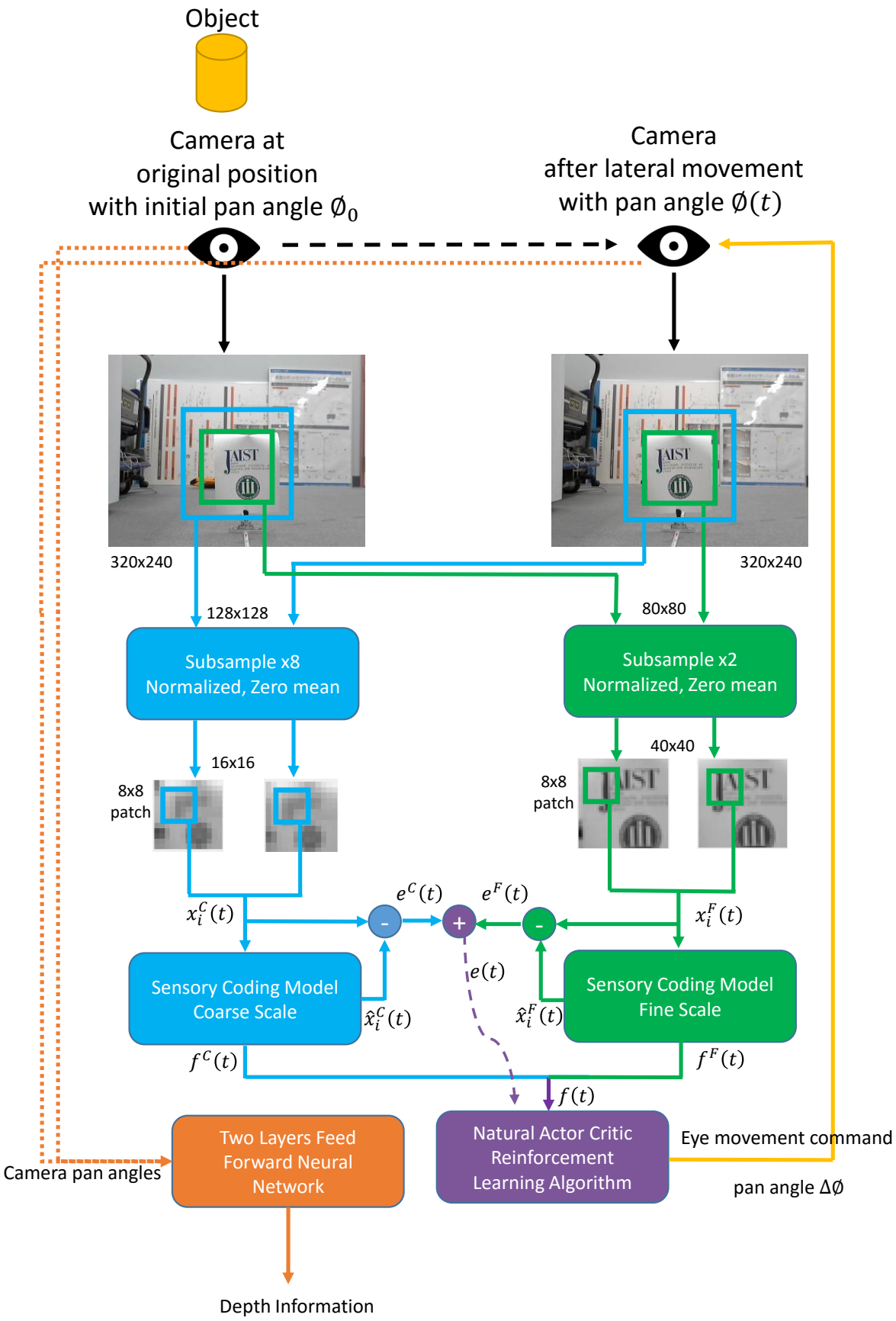


Figure 5.3: Motion parallax framework. Camera is at original position with pan angle initially set to ϕ_0. After lateral movement the camera is panned additionally by $\Delta\phi$ ($\phi(t) = \phi(t-1) + \Delta\phi$) which is a eye movement command received from reinforcement learner part.

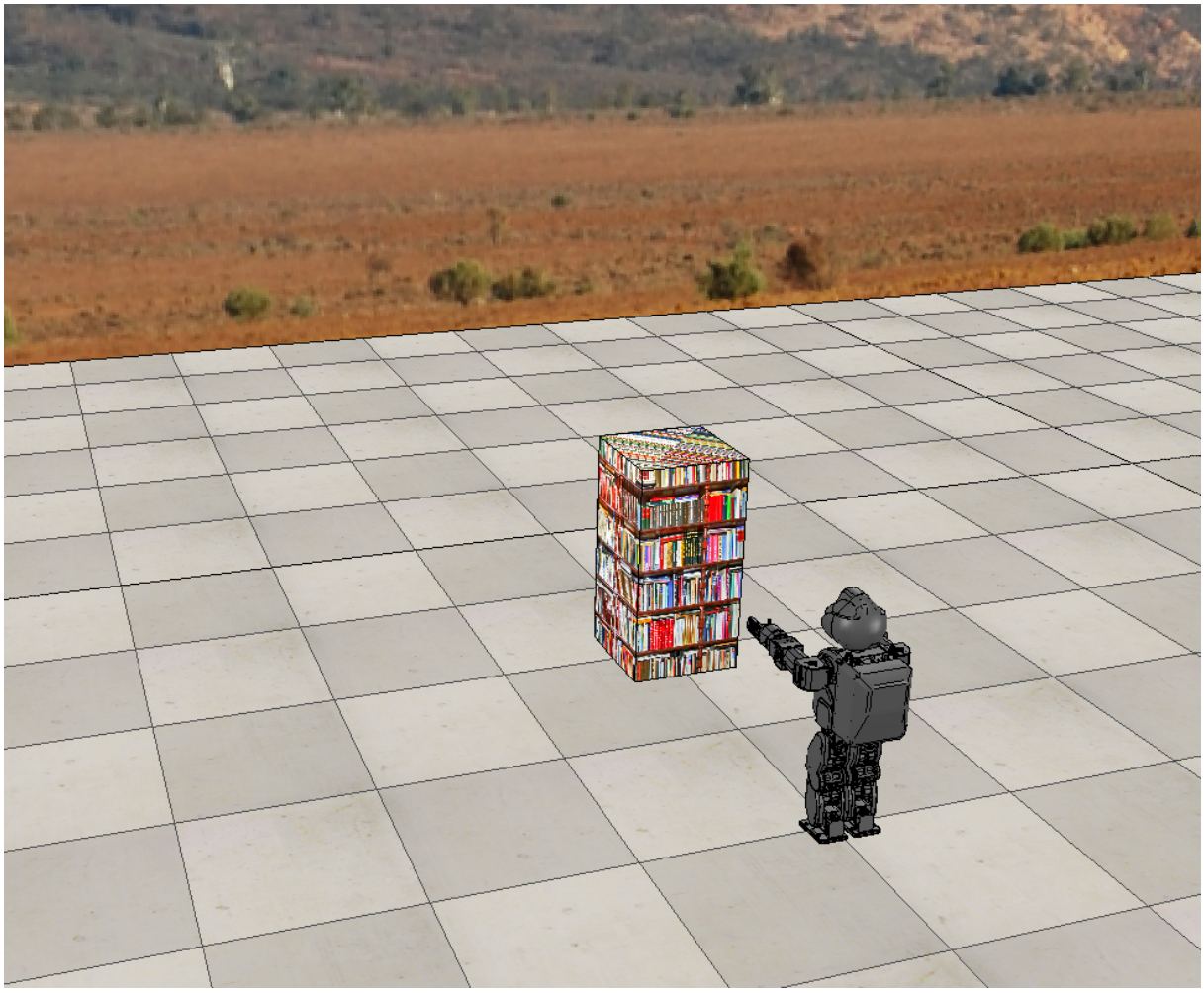


Figure 5.4: Motion parallax framework simulation by using V-REP

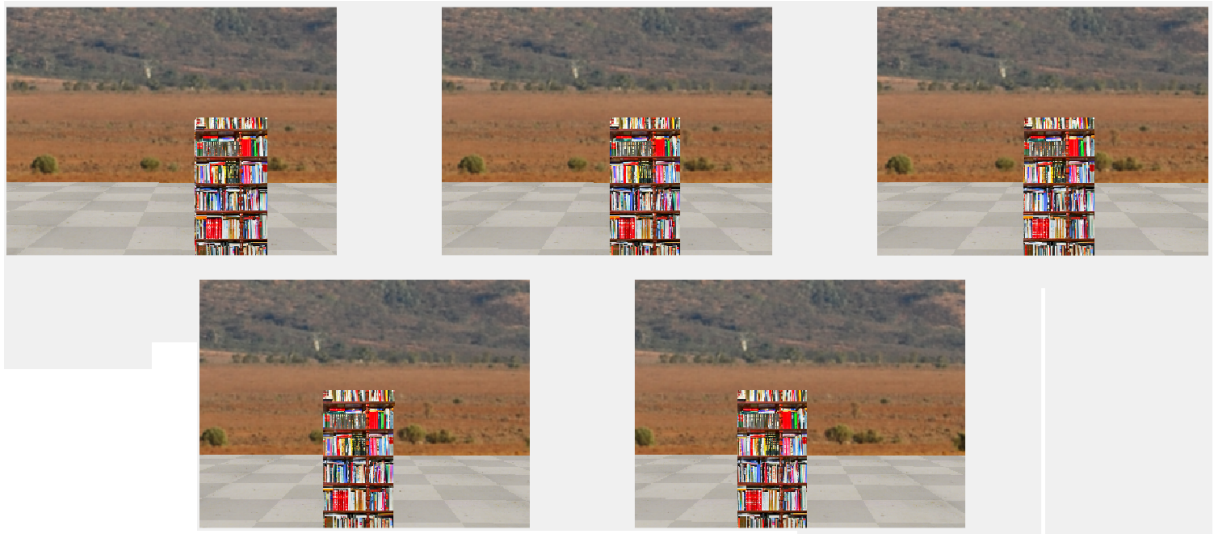


Figure 5.5: Example of motion parallax images from simulation (left to right)

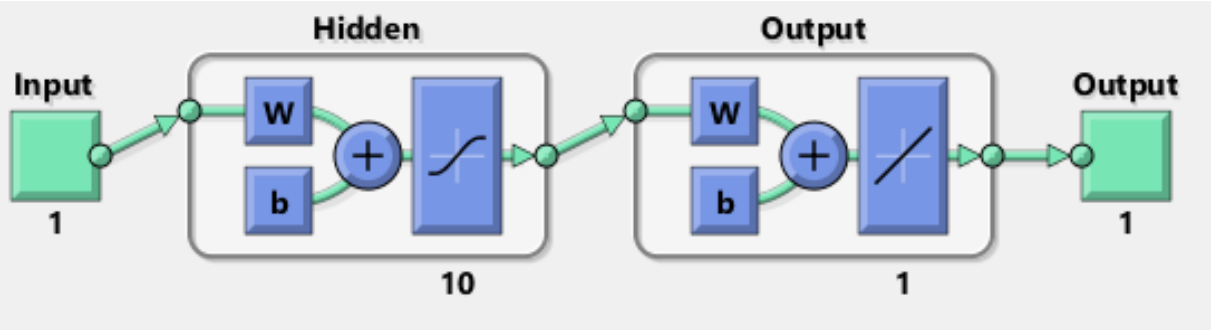


Figure 5.6: The neural network used in this simulation

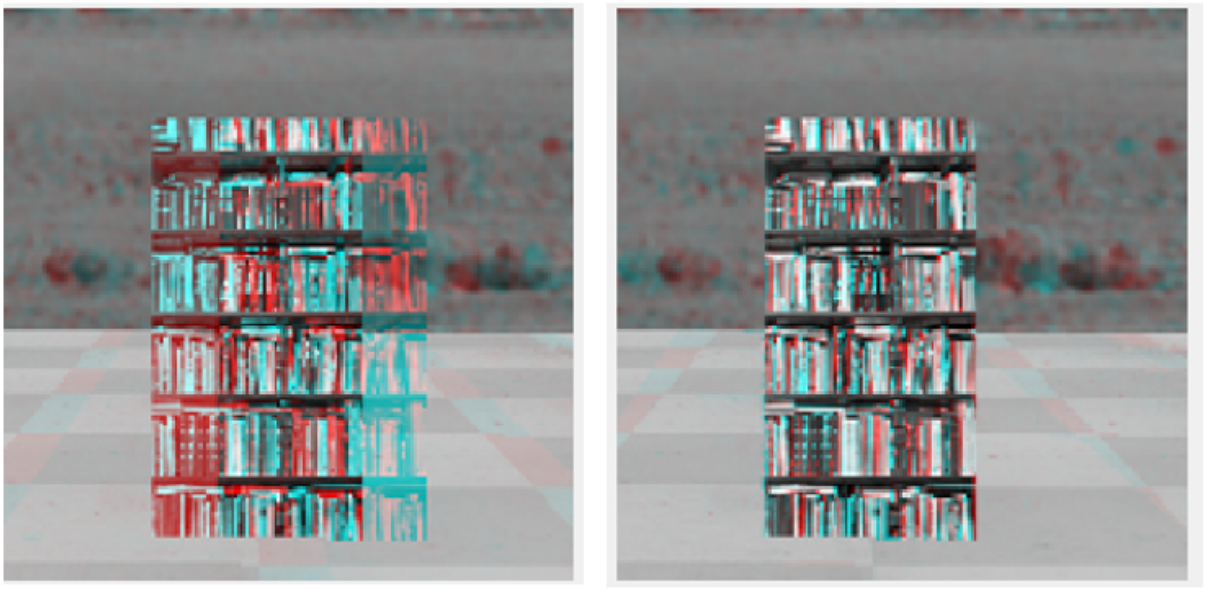


Figure 5.7: Example of object fixating in simulation

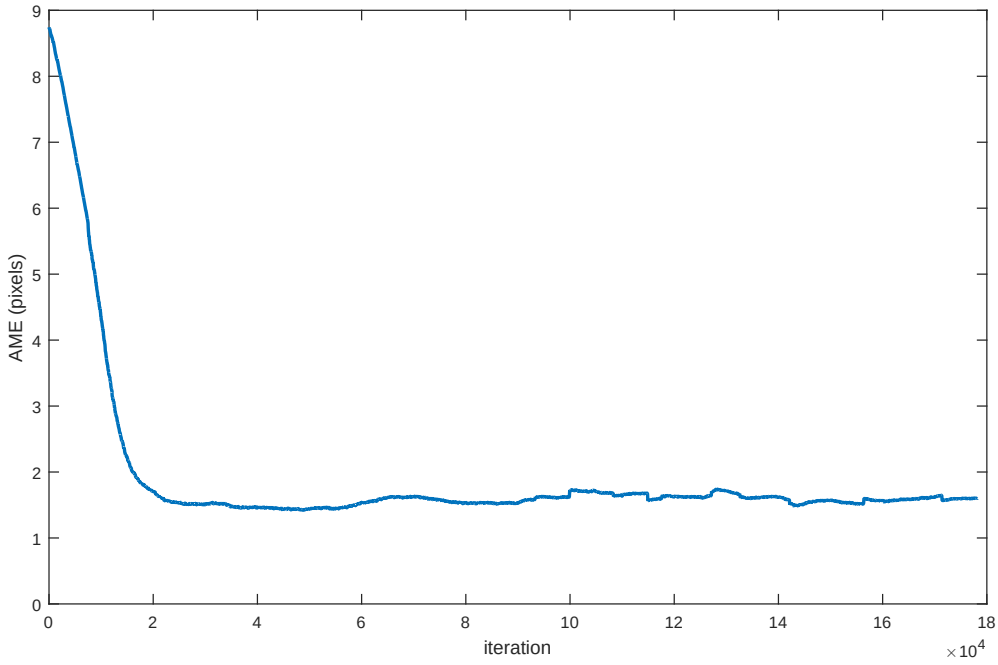


Figure 5.8: MAE of HOAP3 simulation

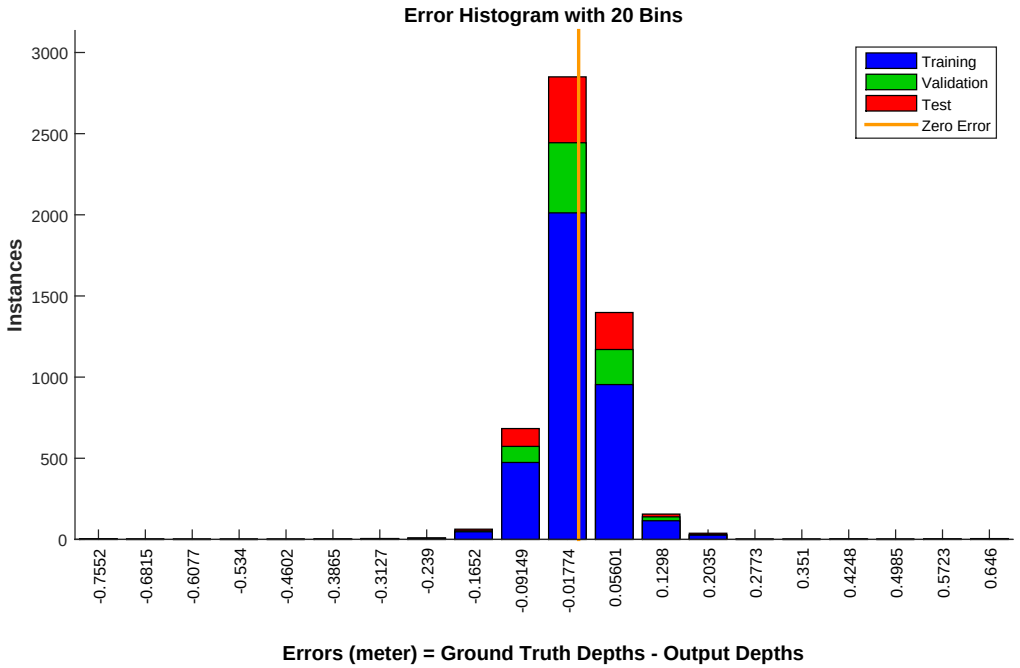


Figure 5.9: Neural network error histogram

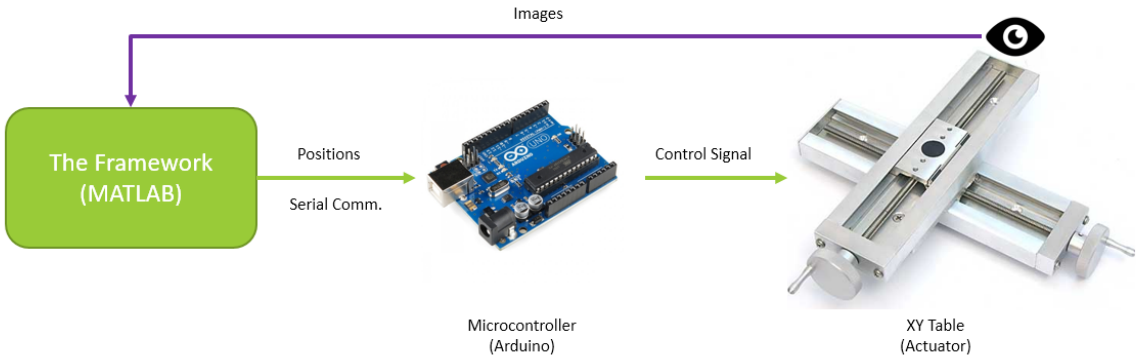


Figure 5.10: Setup for real world experiment

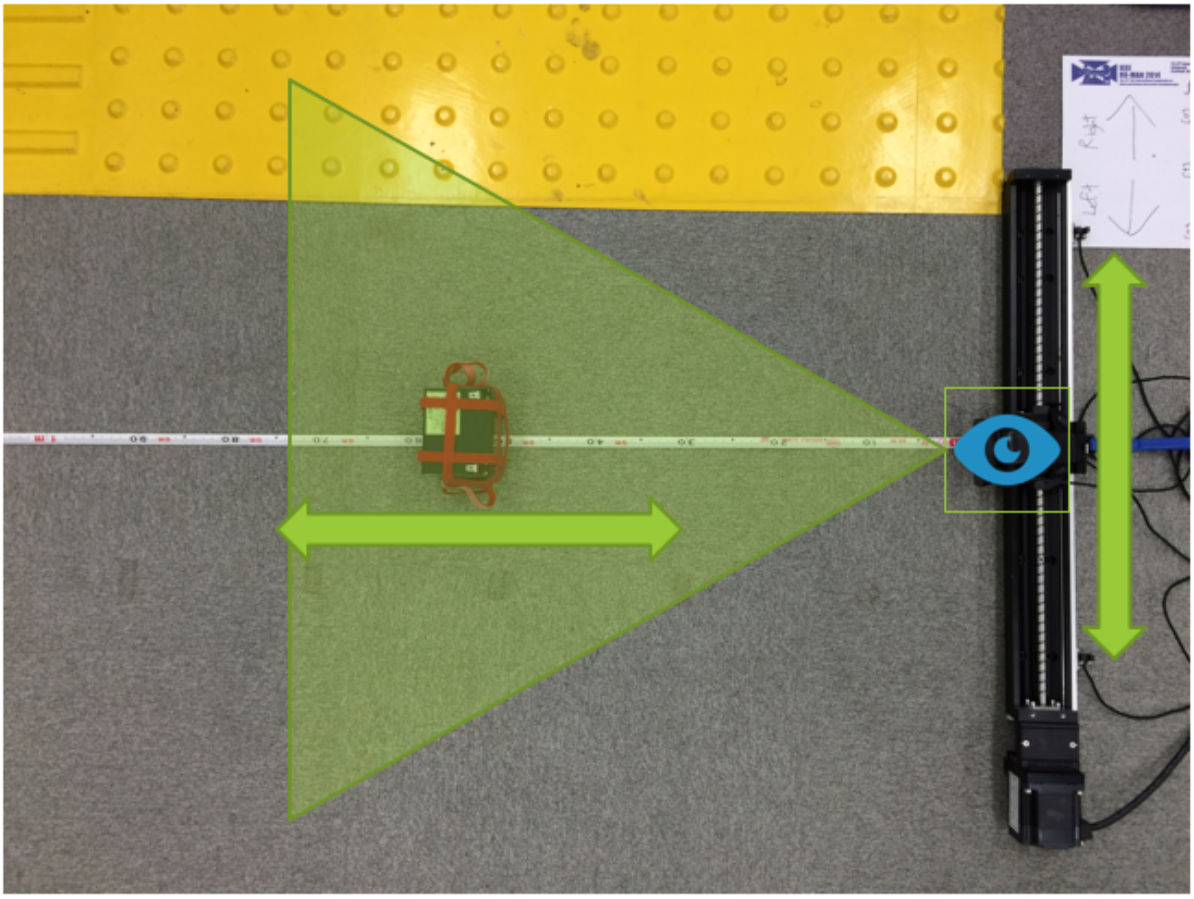


Figure 5.11: XY-table and the object

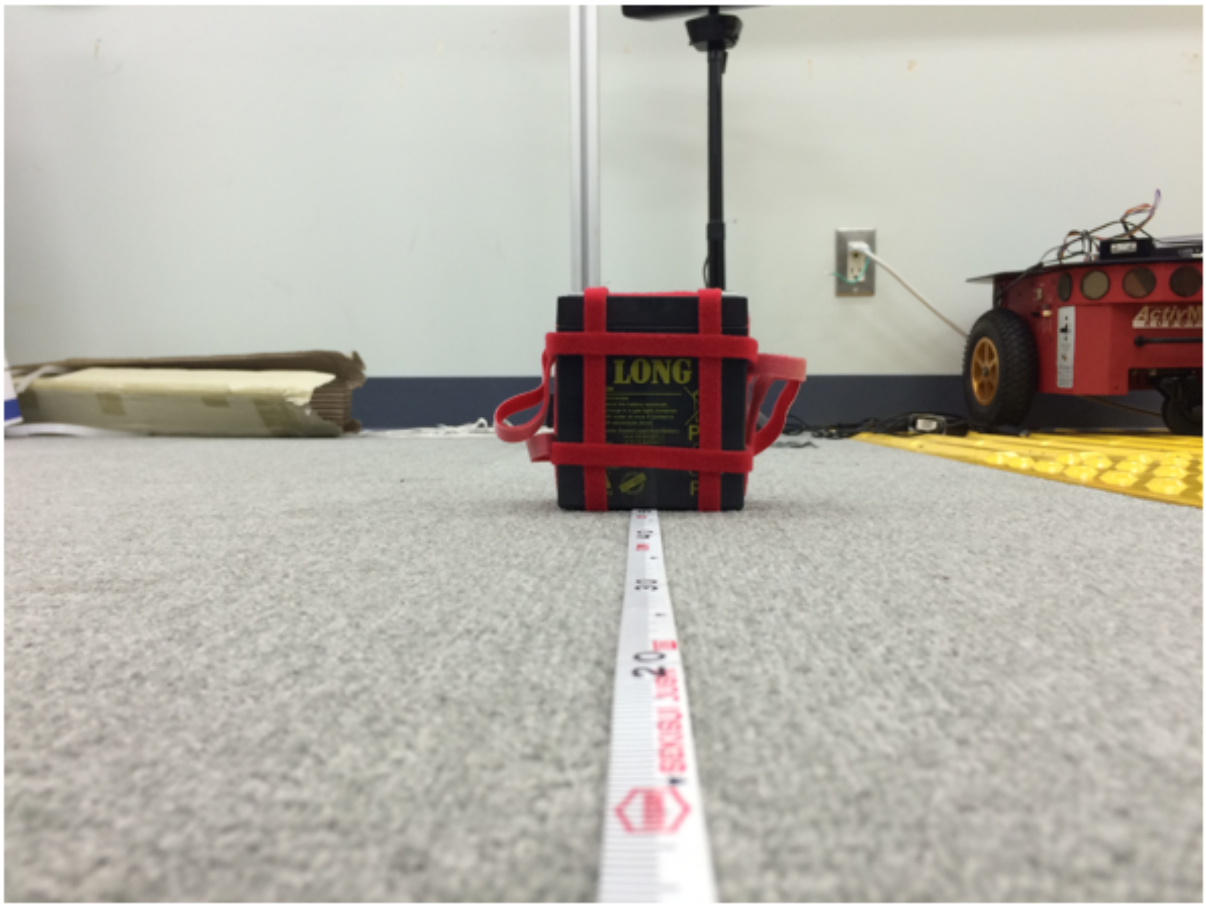


Figure 5.12: View from camera

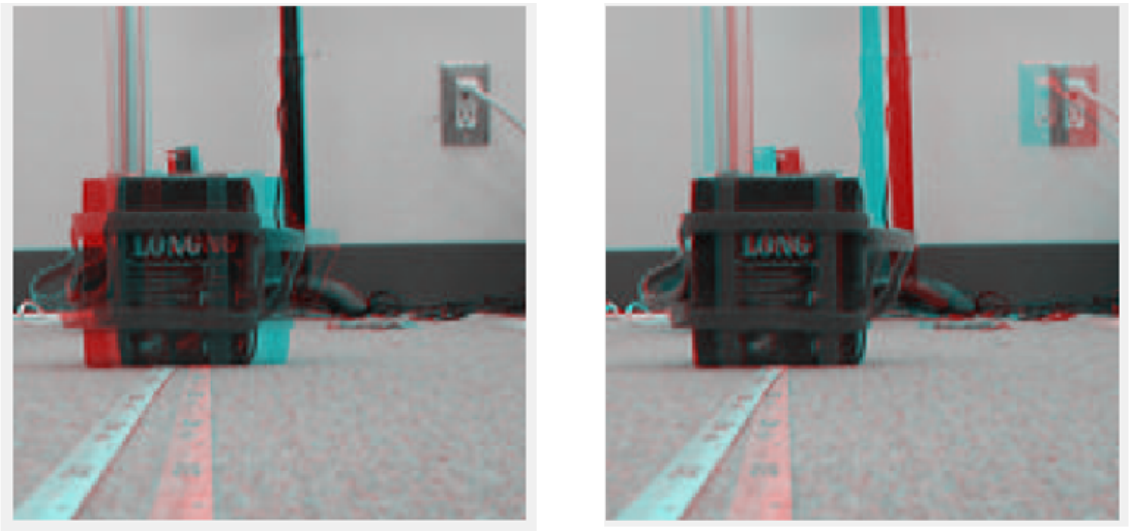


Figure 5.13: Example of object fixating image from real world

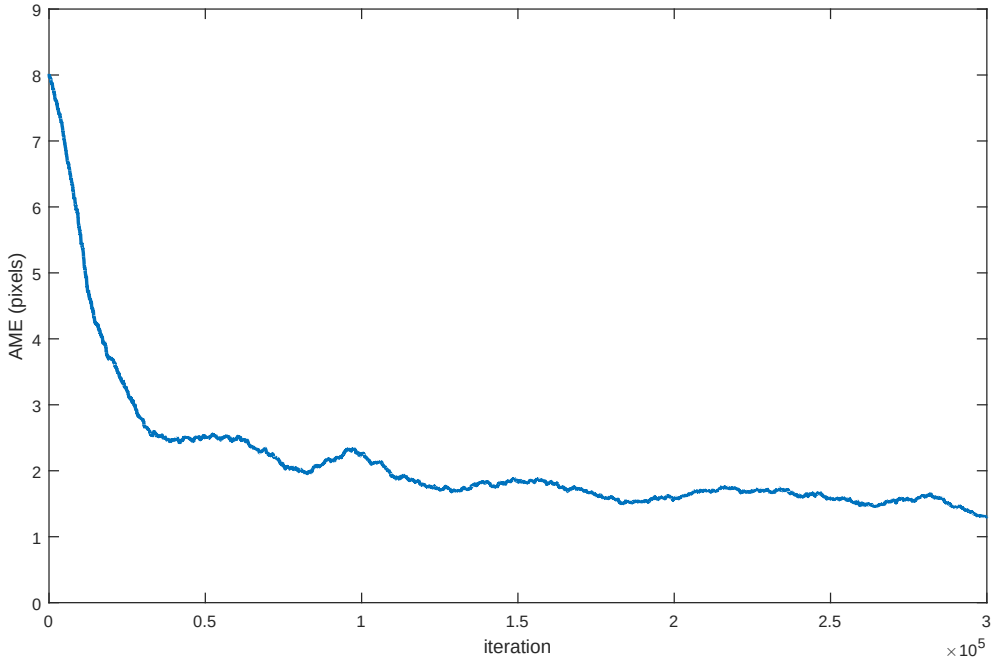


Figure 5.14: MAE of real world experiment

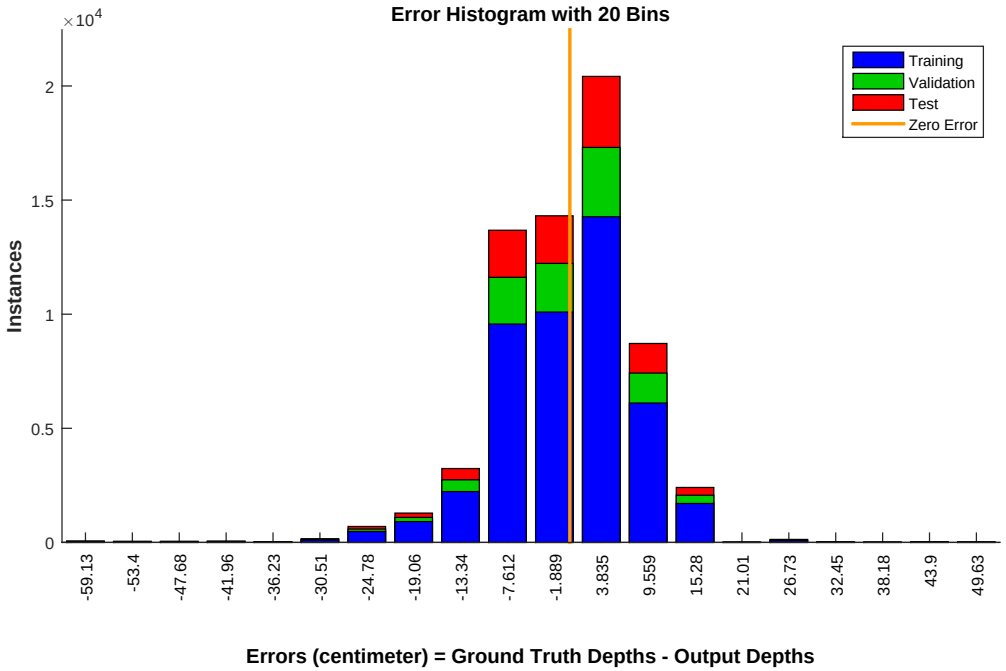
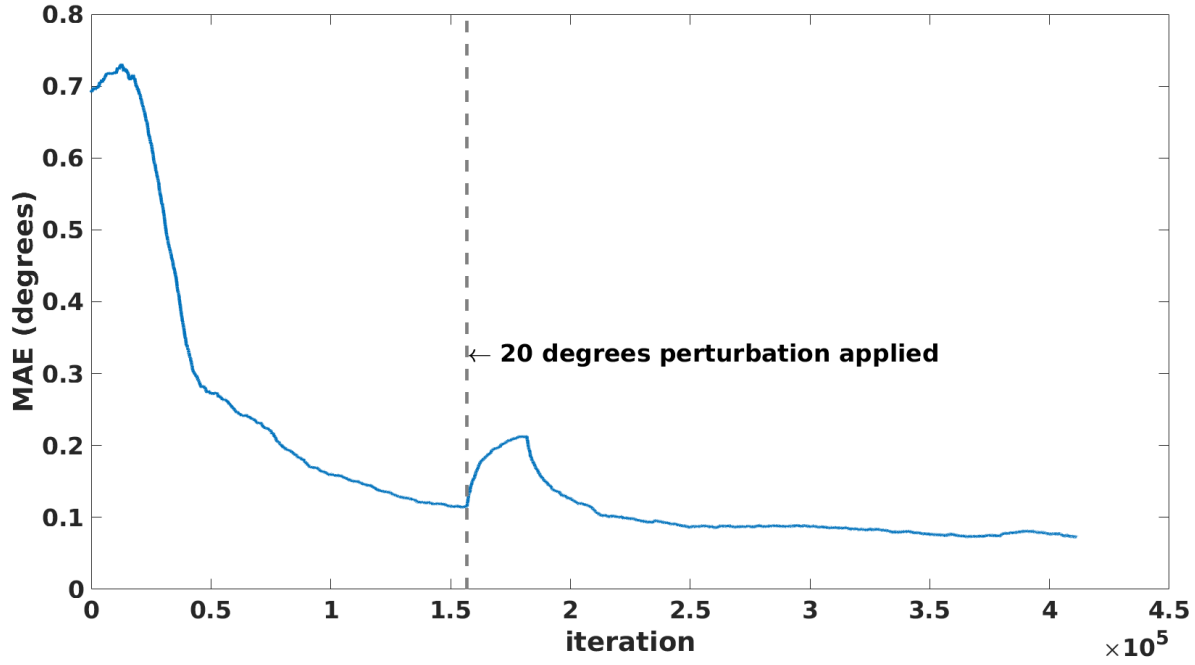
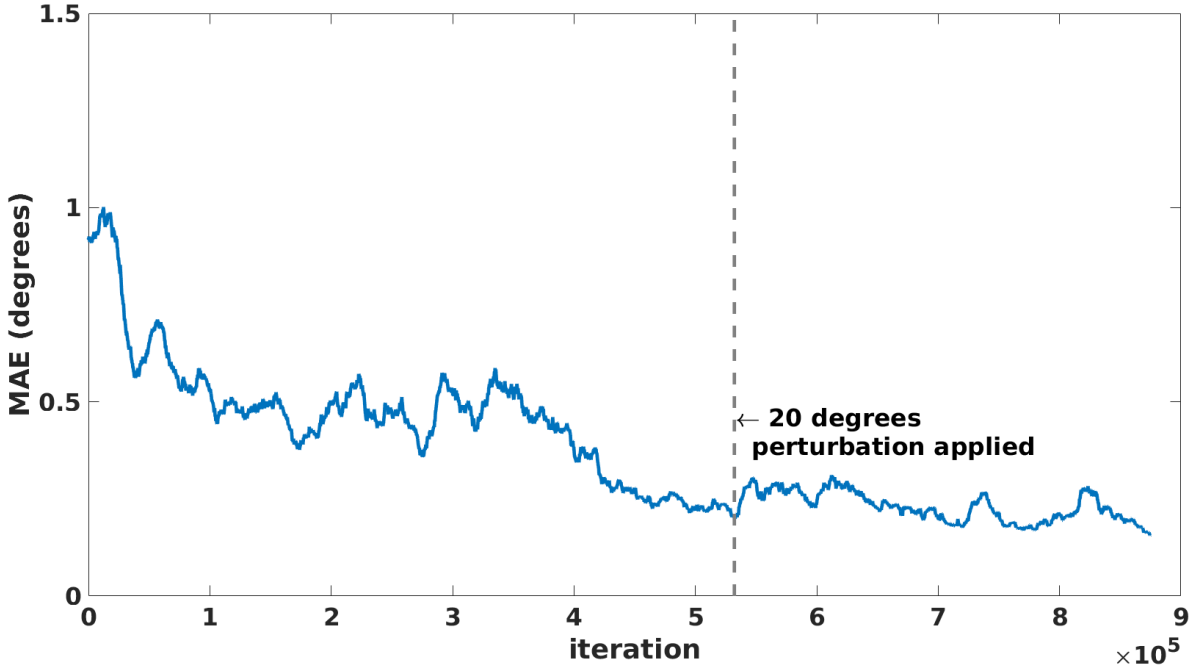


Figure 5.15: Neural network error histogram



(a) Real world experiment



(b) HOAP3 simulation

Figure 5.16: MAE of the simulation and the real world experiment after 20 degrees rotation perturbation

Chapter 6

Schemes of Motion Parallax Based with Multiple Lateral Movement

With the success of extending the framework to be compatible with motion parallax effect, we take a deeper look of how it can be improved. In this chapter, we will discuss how the framework in the chapter 5 can be reinforce with a new learning strategy.

6.1 Philosophy of This Work

When humans move in a lateral direction, they can intuitively understand the motion parallax phenomenon while jointly developing sensory neurons and pursuit eye movements with the help of their life-long learning experiences. At that time, various ranges of the motion parallax are used to extract meaningful pieces of information such as relative depth of variously positioned objects and the spatial separation between the robot and the fixating object (absolute distance).

By mimicking the visual learning in mammals to realize an autonomous robot system, a visual learning framework in the chapter 5 was proposed to concurrently develop both visual sensory coding and pursuit eye movement with an addition of depth perception. Within the proposed framework, an artificial neural network was used to learn the relationship between the eye movements and the absolute distance. Nonetheless, the limitation of the proposed framework is that the predefined single lateral body movement can not fully evoke the motion parallax for depth perception.

In this chapter, we extend the presented visual learning framework to accurately and autonomously represent the various ranges of absolute distance by using the pursuit eye movements from multiple lateral body movements. We show that the proposed model, which is implemented in the HOAP3 humanoid robot simulator, can successfully enhance the smooth pursuit eye movement control with the self-calibrating ability and the distance estimation comparing to the single lateral movement based approach.

6.2 Model Architectures

With monocular vision system, we can utilize motion parallax to perceive depth. The phenomenon is evoked when an observer move laterally while fixating a visual stimulus. By letting a robot move laterally, it can generate the motion parallax effect in many positions and translation speeds. There are two kinds of depth information which are egocentric distance and allocentric distance. In this chapter we focus on only the developments of cognitive functions and the understanding of egocentric distance information.

In this chapter, we try to implement a framework that can generate the smooth pursuit eye movement that can fixate the visual stimulus while the robot is moving laterally at different positions. Fig. 6.1 represents the overview of the model of the framework. The amount of eye movement is used to map the depth information. There are 3 main cognitive functions that is required to be tightly coupled with each other: (1) Visual representation based on sparse coding, (2) eye movements control based on reinforcement learning, (3) artificial networks to represent the distance information by interacting with human.

Sparse coding scheme is used as the sensory coding model which is coupled with the reinforcement learner. With this combination, we can achieve the efficient coding schemes as the sensory coding model. The sensory coding model find the efficient representation of the input images, while the reinforcement learner finds the actions that achieve best image representation. Finally, multiple lateral movements aids the framework to understand various ranges of motion parallax. In addition, the amount of eye movements are input to an artificial neural networks to represent that egocentric distance information.

6.2.1 Single & Multiple Lateral Positions

Two different learning strategies are considered and compared. The strategies are single lateral body movement and multiple lateral body movement. Fig. 6.2 highlights the key difference between the two strategies. With the multiple lateral boyd movement, the robot have multiple difficulty of the learning signal compared to the single lateral movement which has only one single learning difficulty. Fig. 6.2a shows that if the lateral movement was too large then the learning difficulty will also be greatly increased. This is inappropriate for learning at the initial stage. However, for the multiple lateral body movement, Fig. 6.2b, provide gradual steps of learning difficulties which soothe make the learning easier in the earlier state.

Position Setup

Initially, image I_0 is captured from the camera at the original position (home position) for the reference image. After that, the robot moves laterally from the previous position l_0 to l_1 which is selected from $L = \{l_1, l_2, \ldots, l_p, \ldots, l_r\}$. p is the index of the lateral positions on the list L as shown in Fig. 6.3. The framework then proceeds to the next iteration.

Obtaining Motion Parallax

After the robot achieves the position $l = l_1$, Fig. 6.4, the motion parallax effect is evoked. The parallax angle is defined as the angle of difference between the two line of sight which is shown as q in the figure.

An image $I_{l_k}(t)$ is captured from the camera at the lateral position k^{th} to collect the information of the visual stimulus which is then used to generate the smooth pursuit eye movements. The two captured images $I(t) = \begin{bmatrix} I_0 & I_{l_k}(t) \end{bmatrix}$ are input to the sensory encoder to generate one eye movement from the reinforcement learner. This process of capturing $I_{l_k}(t)$ and generating eye movement is repeated for h iterations (one trial). Theoretically, the framework should produce a total amount of eye movements so that it is similar to q.

The robot moves to the next lateral position l_2 which is from L. It then repeats the process until it reaches the last position l_r. After reaching l_r, the robot simply moves back to l_0 preparing for the next visual stimulus.

6.2.2 Sensory Coding Model

Two input images are cropped by 250×250 pixels and 150×150 pixels from the center of the images. Two cropped images represent a fine scale and a coarse scale respectively.

10 by 10 pixels patches are then extracted from the grayscale images, whose locations are generated by 1 pixel and 4 pixels shifts horizontally and vertically for coarse scale and fine scale, respectively. The image patches are sub-sampled, using Gaussian pyramid algorithm by a factor of 8 for coarse scale, and a factor of 2 for fine scale. The patches are reshaped to be one-dimensional vectors which have zero mean and unit norm, $\gamma_i^j(t)$. i is the index of the patch, which $j \in \{C, F\}$. C is for coarse scale, and F stands for fine scale. With the sub-sampled images, the framework will able to handle image disparity that is larger than patch width. Note that the fine scale helps in fine-tuning the eye movements.

For the coarse scale and the fine scale, the two one-dimensional vectors are then combined into a single vector $\gamma^j(t)$. The first 100 elements of the vectors are from the first image I_0 and the remaining are from the second image $I_{l_k}(t)$. The result vectors ($\gamma^C(t)$ and $\gamma^F(t)$) consist of $K = 200$ elements.

Later, the patches are encoded by a sparse coding algorithm in a linear fashion. Each patch is represented by a linear combination of basis functions picked up from a dictionary $\phi^j(t) = \{\phi_n^j(t)\}_{n=1}^N$ [87]. We use $N = 288$ basis functions. Two dictionaries are randomly initialized and normalized. One is for coarse scale and the another is fine scale dictionary as shown in Fig. 6.1.

For the coarse scale and the fine scale, the two one-dimensional vectors are then combined into a single vector $\gamma^j(t)$. The first 100 elements of the vectors are from the first image I_0 and the remaining are from the second image $I_{l_k}(t)$. The result vectors ($\gamma^C(t)$ and $\gamma^F(t)$) consist of $K = 200$ elements.

Later, the patches are encoded by a sparse coding algorithm in a linear fashion. Each patch is represented by a linear combination of basis functions picked up from a dictionary $\phi^j(t) = \{\phi_n^j(t)\}_{n=1}^N$ [87]. We use $N = 288$ basis functions. Two dictionaries are randomly initialized and normalized. One is for coarse scale and the another is fine scale dictionary as shown in Fig. 6.1.

We use the matching pursuit algorithm to estimate and find the sparse representation

of the input vector by the weighted sum as follows:

$$\gamma_i^j(t) \approx \hat{\gamma}_i^j(t) = \sum_{n=1}^{N} b_{i,n}^j(t)\phi_n^j(t) \ . \tag{6.1}$$

The matching pursuit algorithm suits to the concept of sparse coding, because it can estimate $\Gamma_i(t)$ by using a limited number of coefficients. In this research, the maximum number of non-zero scalar coefficients $b_{i,n}(t)$ is set to be 10 elements to ensure sparseness of the efficient coding. For later use in reinforcement learner part, pooled activity, $\theta_n(t)$, which represent the activity of each neuron cell is calculated from the coefficients from the matching pursuit algorithm as follows:

$$\theta^j(t) = \begin{bmatrix} \theta_1^j(t) \\ \theta_2^j(t) \\ \vdots \\ \theta_N^j(t) \end{bmatrix} \ . \tag{6.2}$$

Where, each element of the vector $\theta^j(t)$ is described as:

$$\theta_n^j(t) = \frac{1}{P}\sum_{i=1}^{P} b_{i,n}^j(t)^2 \tag{6.3}$$

, where P is the number of patches extracted from one input image. A reconstruction error is introduced as a cost function to be used in sensory coding model and reinforcement learner. It measures the estimation error of vector $x(t)$. The reconstruction error is defined as:

$$e^j(t) = \frac{1}{P}\sum_{i=1}^{P} \frac{\|\gamma_i^j(t) - \sum_{n=1}^{N} b_{i,n}^j(t)\phi_n^j(t)\|^2}{\|\gamma_i^j(t)^2\|}. \tag{6.4}$$

A gradient descent method is used to update the dictionaries with the reconstruction error as a cost function. After each update, the dictionaries are then normalized.

6.2.3 Reinforcement Learning

The state representation of the reinforcement learner can be described by combination of coarse scale and fine scale pooled activity, $\theta_n(t)$ as follows:

$$\theta(t) = \begin{bmatrix} \theta^C(t) \\ \theta^F(t) \end{bmatrix} \ . \tag{6.5}$$

The reward that is given to the learning agent is a negative of the summation of reconstruction error from both scales which is described as:

$$R(t) = -(e^C(t) + e^F(t)) \ . \tag{6.6}$$

The actor-critic algorithm number 3 proposed in [95] is employed for the leaner agent. For action selection, we use Gibbs distribution (softmax) for probabilistically choosing an action as follows:

$$\pi(\theta(t), a_t) = \frac{e^{z_a}}{\sum_{a' \in A} e^{z_{a'}}} \ . \tag{6.7}$$

For each action, the activation value z_a is given by:

$$z_a = \sum_{n=1}^{N} w_a(t)\theta_n(t) \ , \tag{6.8}$$

where $w_a(t)$ is a weight vector from the state $f(t)$ to action a. The action is pan angle of the cameras in degrees. Possible actions a are contained in a set of actions A. In this research we use $A = \{-0.2°, -0.1°, -0.05°, 0°, 0.05°, 0.1°, 0.2°\}$. Thus, the policy maps $\theta(t)$ to $a \in A$.

6.2.4 Depth Representation

To extract the distance information, one may calculate it directly with the knowledge it has, such as traveled distance and eye movement. Since in this research, we focus on building the framework that can adapt to the various configuration of the system and environments, so it is impractical to specifically calculate the distance information which usually requires exact system's parameters.

To let the system learn the relationship between the distance and the eye movements, the robot must know (1) lateral distance and (2) amount of eye movements. Since the robot moves according to the lateral position list L, the speed of the lateral translation is constant. So, the knowledge of the lateral distance can be excluded from the learning. For the amount of eye movements, at the end of each trial of each lateral position, the eye movements (q in Fig.6.3) are memorized and accumulated in the vector $\vec{q} = \begin{bmatrix} q_1 & q_2 & \dots & q_p & \dots & q_r \end{bmatrix}^\top$.

We suppose that the robot can remember c of its previous eye movement vector $\vec{q}$. The previous eye movement vectors are concatenated to create a queue-memory matrix

$$Q = \begin{bmatrix} \vec{q_1} & \vec{q_2} & \cdots & \vec{q_c} \end{bmatrix} \tag{6.9}$$

, where $\vec{q_1}$ is the latest eye movement vector collected, and $\vec{q_c}$ is the last eye movement vector that the robot can remember. When a new eye movement information $\vec{q}$ is available, $vecq_c$ in the matrix is discarded (dequeued). The indexes of the remaining vectors are shifted by one, i.e. $\vec{q_k}$ is assigned to be $\vec{q_{k+1}}$. The new vector is then assigned (queued) as $\vec{q_1}$

Here, we use a feed-forward neural network with a hidden layer as the egocentric distance learner to interpret the eye movements to the distance information. We use Levenberg-Marquardt method [102] for training the neural network. The input of the neural network is Q (batch training). The sigmoid activation function is used in the hidden layer which has 30 neurons. The output layer uses the linear activation function. The target is ground truth distances provided by the supervisor. The supervised signals (ground truth absolute distances) are provided for letting the robot understand the metric system. The neural network starts to train after the robot has filled the memory matrix Q, i.e. q_c exists. The training occurs every iteration that the new $\vec{q}$ is available.

Normalization of Generated Eye Movements for Neural Networks

Sensitive information should be carefully used as inputs for neural networks. Because large lateral body movement makes the eye movement generation more difficult, the generated eye movement from large lateral body movements that would be considered sensitive information.

To determine which lateral body movement gives sensitive to eye movements, we consider disparity scores which tell how much the sensory encoder can derive the information from the two input images. The lateral body movements that give disparity score, which is higher than the patch size, are marked as sensitive lateral positions. We try to reduce the negative effect of the sensitive information by weighting the neural network input Q.

If the lateral position l_p is the considered as the sensitive lateral position, then the lateral positions l_p to l_r are sensitive. Weighting is then applied to the inputs that are from the sensitive lateral positions which can be expressed as follows:

$$Q = \begin{bmatrix} w_1 q_{1,1} & w_1 q_{1,2} & & w_1 q_{1,c} \\ w_2 q_{2,1} & w_1 q_{2,2} & & w_1 q_{2,c} \\ \vdots & \vdots & & \\ w_p q_{p,1} & w_p q_{p,2} & \cdots & w_1 q_{p,c} \\ \vdots & \vdots & & \\ w_r q_{r,1} & w_r q_{r,2} & & w_1 q_{r,c} \end{bmatrix} \tag{6.10}$$

. The weight w_k is defined as:

$$w_k = \begin{cases} \dfrac{1}{(1 + l_r - l_k)} \cdot \dfrac{1}{y}, & \text{if } k \geq p \\ 1, & \text{otherwise} \end{cases} \tag{6.11}$$

where y is a hyperparameter.

6.3 Simulations & Results

6.3.1 Experimental Setup

V-REP is used as the environment simulator for the robot as shown in the top picture in , while MATLAB is used to implemented the framework Fig. 6.1. In this simulation the egocentric distance can be varied in 0.1 meters interval from 3 meters to 10 meters. The object's texture is interchangeable with the prepared 100 images for learning of the various texture in environment.

Joint Development of Active Depth Perception

In this section, we analyze and verify the performance of the system. $c = 300$ eye movements are used as inputs for the neural networks to test the distance estimation. Eye movement generation and reconstruction errors are observed to verify the progress of learning. To track the performance of the eye movement generation, eye movements at 30th iteration are recorded and compared with the expected eye movement. The mean absolute error (MAE) is computed to evaluate the training.

In order to compare the two different learning strategies, we test the performance of the framework with 6 different set of experiments. Each set of the experiments use different

Table 6.1: Disparity score of the two input images at the beginning of each trial at 3 meters distance

Lateral Position (cm)	Disparity (pixel) at 3 m
5	9
7	12
10	18
13	23
15	27
20	35

single lateral position. We use $L = 5, 7, 10, 13, 15$ and 20 cm as the lateral positions. Every set of the experiments contains 3 simulations.

We picked these lateral positions based on the disparity score which is an approximated distance between the two input images in pixels, as seen in Table 6.1 below. The disparity score is calculated by using geometry at 3 meters distance, because at the closest distance we can observe the maximum disparity for each lateral position.

Because the coarse scale is sub-sampled from the original image by 8, the maximum horizontal disparity that causes no overlap between two 10x10 pixels patches would be $10 \cdot \sqrt{8}$ which is around 28. At 15 cm lateral position, it barely gets two patches overlap at the start of each trial. While 20 cm lateral position has no overlap. These 2 lateral positions would be good examples of the effect of having large lateral movement.

After we confirmed the training of the single lateral position simulation, we test the performance of the system with two sets of multiple lateral positions which are $L = \{5, 6, 7, \ldots, 10\}$ (cm) and $L = \{5, 6, 7, \ldots, 20\}$ (cm).

6.3.2 Performance Comparison

The training results of multiple lateral positions are shown in Fig. 6.12 for 5-10 cm and Fig. 6.13 for 5-20 cm, respectively. The blue dashed lines represents the variance of each trial from 3 simulations. The solid line is the average of the MAE from 3 simulations. Table 6.2 show the comparison results between a single lateral body movement and multiple lateral movements.

Table 6.2: Performance of the single lateral position (Sing.) and multiple lateral positions (Mult.)

Lateral Position	Average (Variance) of last 100 trials		
	Sing.	Mult. 5-10 cm	Mult. 5-20 cm
5	0.18 (0.10)	0.18 (0.08)	0.25 (0.13)
7	0.16 (0.06)	0.16 (0.06)	0.18 (0.06)
10	0.16 (0.02)	0.17 (0.06)	0.18 (0.07)
13	0.18 (0.02)	-	0.17 (0.03)
15	0.30 (0.08)	-	0.17 (0.03)
20	0.91 (1.15)	-	0.19 (0.04)

Some of the lateral position has a high peak of MAE and is slower to get stable, because the redundancies between input images is very low initially, so it is quite difficult for the framework to learn with such small information.

As shown in Table 6.2, all of the simulation except for the 15 and 20 cm lateral position show a similar performance in terms of the last 100 trials eye movement MAE. In the beginning period of training, there are a lot of combinations of texture and distance of the object to be learned, so the rises and declines of the MAE are expected as seen in Fig. 6.21. When the lateral movement is too large, it makes the distance between two input images I_1 and $I_2(t)$ initially large. So, the framework could not utilize the redundancy between the two images effectively, which results in unstable eye movement generation as shown in Fig 6.19. However, we can see that it can still maintain the eye movement with enough precision.

6.3.3 Robustness Test

To test the robustness of the system, we apply two types of perturbation to the robot. First, the camera is rotated by 15 degrees in roll-plane. Second, a Gaussian filter with standard deviation of 2 is applied to the input image to represent the blurriness.

The disturbances are applied after the training which are shown in sub-section 6.3.2. As shown in Fig. 6.14 – Fig. 6.21 and Table 6.3, noticeable increasing in the MAE are observed after including the disturbance which is presented by gray dashed line in the

Table 6.3: Performance of the single lateral position (Sing.) and multiple lateral positions (Mult.) after perturbations applied.

Lateral Position	Average (Variance) of last 100 trials		
	Sing.	Mult. 5-10 cm	Mult. 5-20 cm
5	0.12 (0.01)	0.13 (0.01)	0.16 (0.02)
7	0.14 (0.01)	0.15 (0.01)	0.16 (0.03)
10	0.21 (0.04)	0.18 (0.02)	0.18 (0.01)
13	0.36 (0.12)	-	0.21 (0.04)
15	0.54 (0.21)	-	0.25 (0.04)
20	1.09 (1.03)	-	0.39 (0.13)

figure.

For small single lateral positions, the framework can recover from the disturbances to have a similar performance before the interferences. For the larger lateral positions from 10 cm, we can see that they could not recover MAE to be similar to the performance before the perturbations. However, they could recover and maintain the MAE. Interestingly, for the 20 cm lateral position test, it can still maintain the MAE as shown in Fig. 6.19.

Noticeably, the lateral positions from 5 to 10 cm can fully recover to the similar or even better performance from the disturbances. Some perform better after the disturbances because it simply has more time to learn. Also, disturbances encourage the framework to explore and learn more. While at the lateral positions form 13 to 20 cm, the framework could not fully recover from the disturbances. Because, combining with high disparity scores and the disturbances, the framework could not effectively learn to generate eye movement. However, it still shows that it can maintain the MAE.

6.3.4 Distance Estimation

Without the weighting algorithm, the depth representation is still useful for the far distance such as 5m to 10m, but the closer distance such as 3m to 5m is worse. With the proposed weighting, it is shown that the depth estimation is improved for both near and far distance. In addition, the proposed model is robust to the perturbations.

The performance and robustness of the distance estimation is investigated in Fig 6.22

Table 6.4: Average distance estimation error for each range of distances.

Simulation	3 to 4.9 (m) distances	5 to 6.9 (m) distances	7 to 10 (m) distances	Total Average
5-10 cm without weighting	3.97%	4.09%	4.74%	4.33%
5-20 cm without weighting	6.39%	2.24%	3.06%	3.77%
5-20 cm with weighting	**3.66%**	**1.89%**	**2.55%**	**2.69%**

Table 6.5: Average distance estimation error after perturbations for each range of distances.

Simulation	3 to 4.9 (m) distances	5 to 6.9 (m) distances	7 to 10 (m) distances	Total Average
5-10 cm without weighting	3.95%	3.52%	5.50%	4.48%
5-20 cm without weighting	4.30%	2.76%	3.03%	3.31%
5-20 cm with weighting	**2.29%**	**0.98%**	**0.82%**	**1.30%**

and Fig. 6.23, respectively. The distance estimation performances are shown. Minimum and maximum error of single lateral movement distance estimation are represented by the blue solid line. The red solid line shows the performance of the multiple lateral positions.

We can see in Fig. 6.22 that by applying multiple lateral position learning strategy is better compared to the average and the minimum of the single lateral position. With the weighting shown in red line, performance is significantly improved at 3-5m distance range. In addition, comparing to the magenta line (without weighting), the proposed learning strategy has better overall performance in far distance 9-10m.

In Fig. 6.23, it shows the performance after the disturbance which is similar to the before interruption for both single and multiple lateral position. Table 6.4 and Table 6.5 show the distance estimation error in each distance. The lateral position from 5 to 20cm with weighting performs better than the other two strategies in every case. In addition, the performance of every lateral movement strategies is robust to the perturbations. We can say that the proposed learning scheme is robust to the changing of the system's parameters.

6.4 Summary

In this chapter, we propose a novel visual learning framework to actively perceive the various ranges of distance from motion parallax by integrative learning of sensory rep-

resentation and eye pursuit during self-induced multiple lateral body movements. An artificial neural network was used to represent the egocentric distance by autonomously understanding the relationship between the amount of eye movements and distance information under a human supervision instead of a certain equation. The generated multiple eye movements were effectively used to represent the distance information and it has a better accuracy to perceive the distance than a single body movement. Moreover, the proposed model also can seamlessly recover the artifacts from the perturbations such as image blur and rotation.

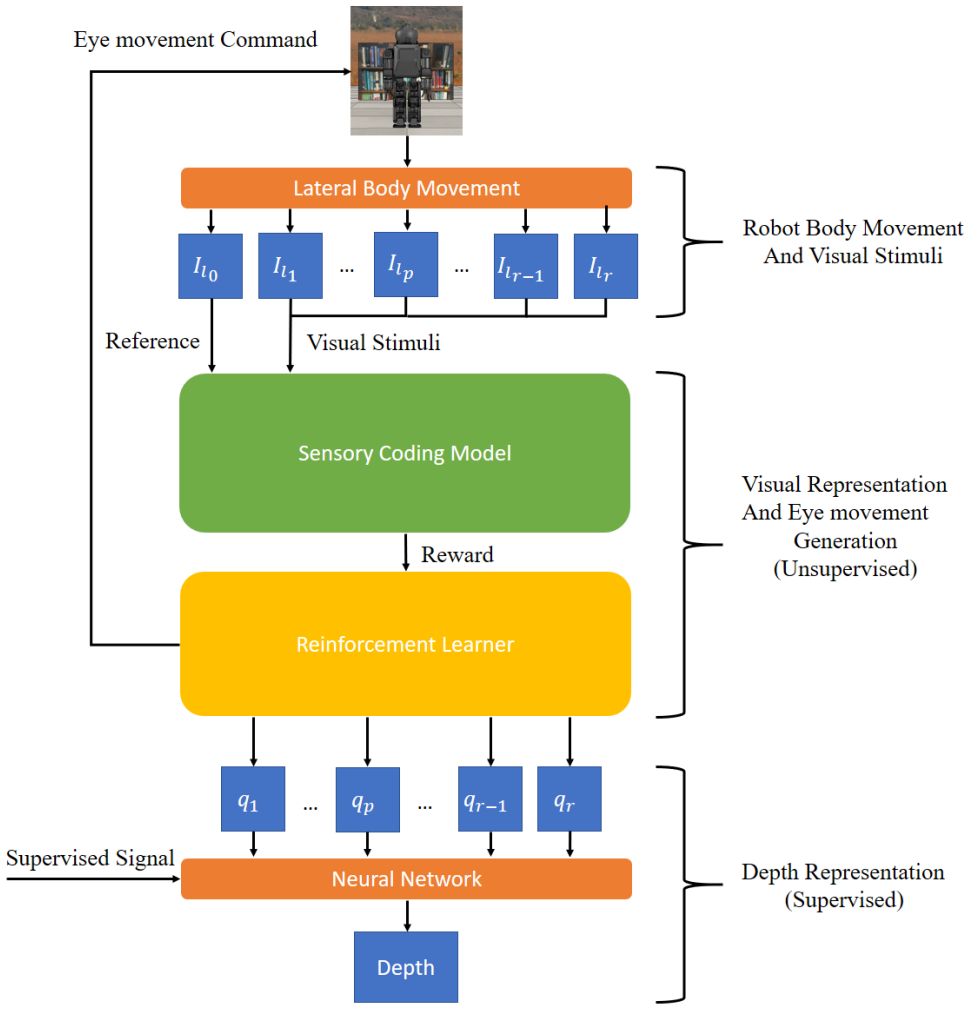
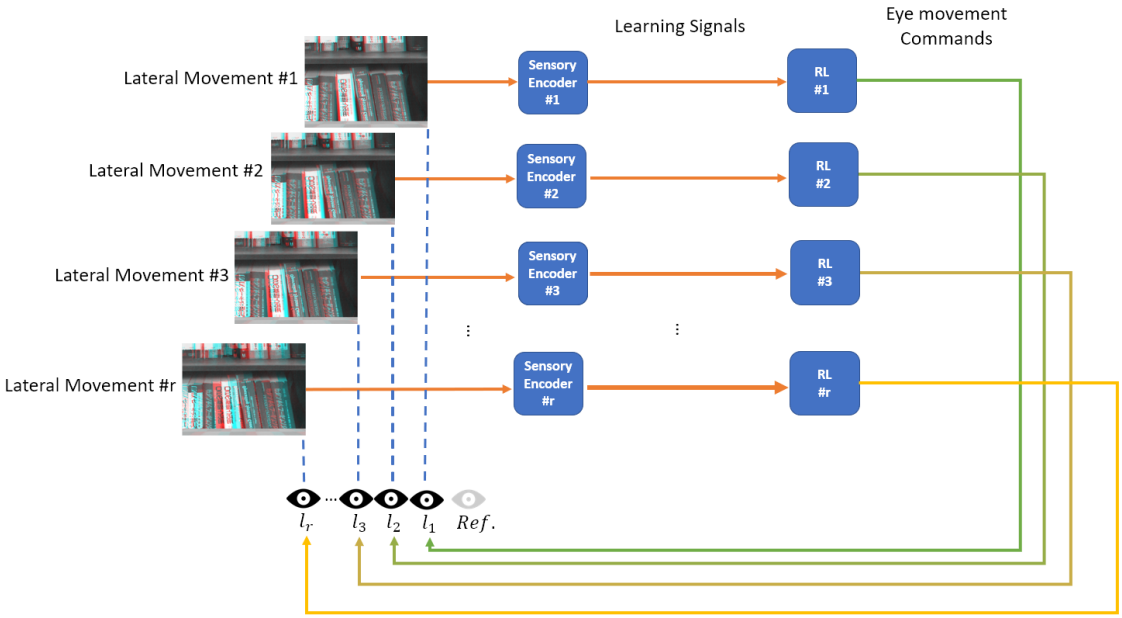
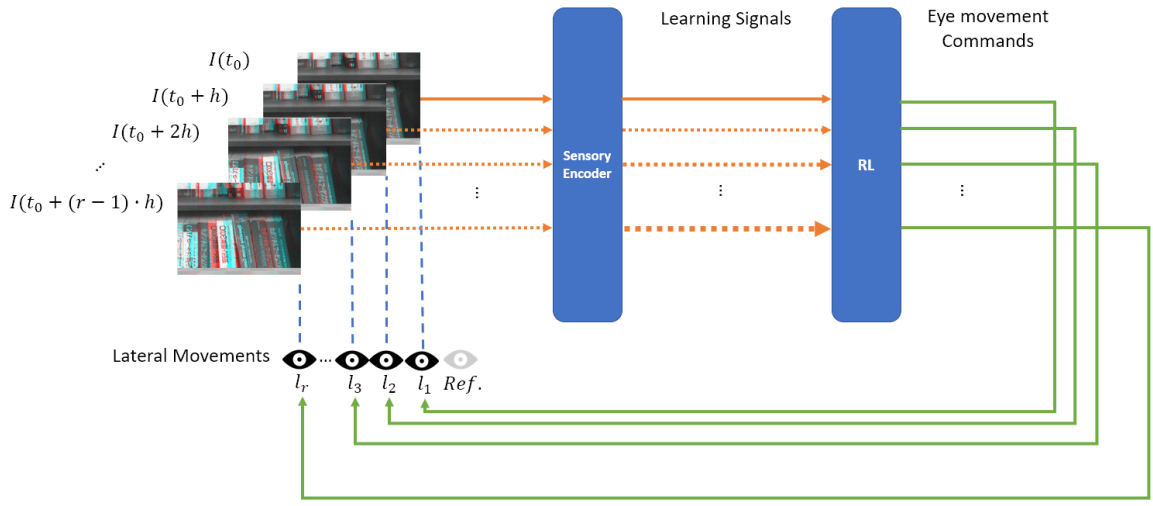


Figure 6.1: Model architecture. The robot captures a reference image and then moves to the lateral position l_k from L. To perform the motion parallax, the successive images $I(t)$ into the sensory encoders with multiple image scales. Then, an output reward signal generated from the sensory encoders is sent to the reinforcement learner to generate an appropriate eye movement to hold the fixation during the body movement. Finally, a pan command is sent to the robot and it generates the smooth pursuit eye movement to maximize the redundancy between the successive images. The memorized eye movements $(q_1, q_2, \ldots, q_r)$ are used as an input for the neural network to represent the distance information which is given by human-robot interaction.



(a) Single lateral distance



(b) Multiple lateral distance

Figure 6.2: a shows a learning scheme when using only single lateral movement. It has only one scale of learning signal. While, b shows the flow of performing the same task but with multiple lateral body movement. It can provide multiple scale of learning signal to the reinforcement learner.

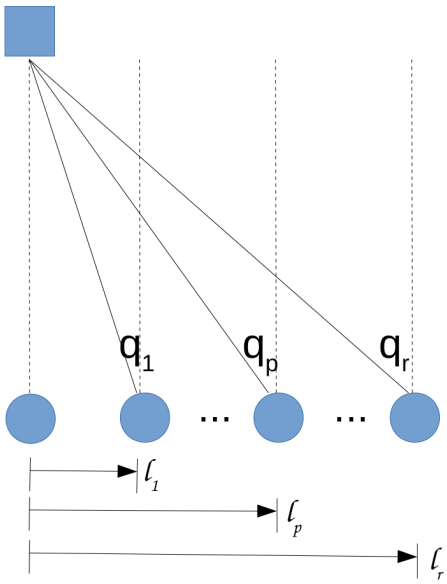


Figure 6.3: The lateral body movement of the robot and the total eye movements at each position. The robot moves laterally for a certain distance from L. Then it tries to generate eye movements $q_1, q_2, \cdots, q_p, \cdots, q_r$ to fixate the visual stimulus at the center of the gaze.

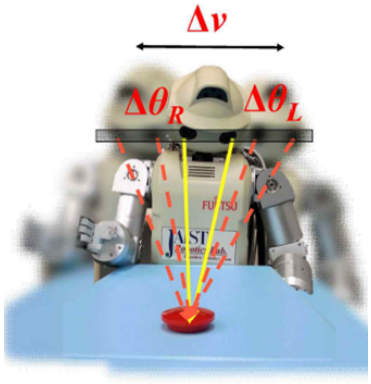


Figure 6.4: The parallax angle q which is identical to the total eye movement required to fixate the stimulus at a certain lateral distance l.

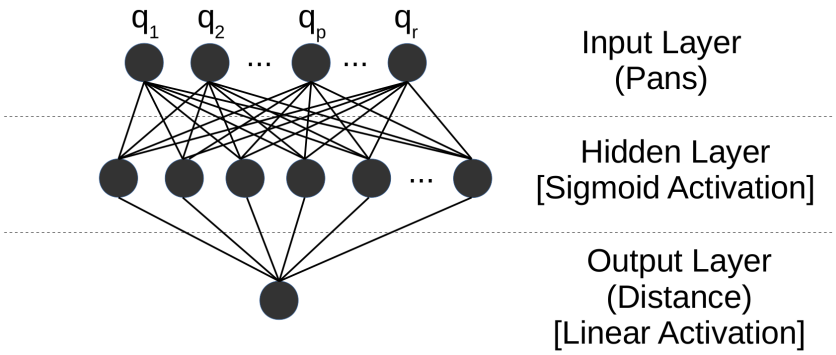


Figure 6.5: The 3 layers feed forward neural network for estimating the egocentric distance. The feature inputs are the eye movements from each lateral position in L. Sigmoid activation function is used in the hidden layer, while the output layer uses linear activation function. The output layer has only one node which is the absolute distance.

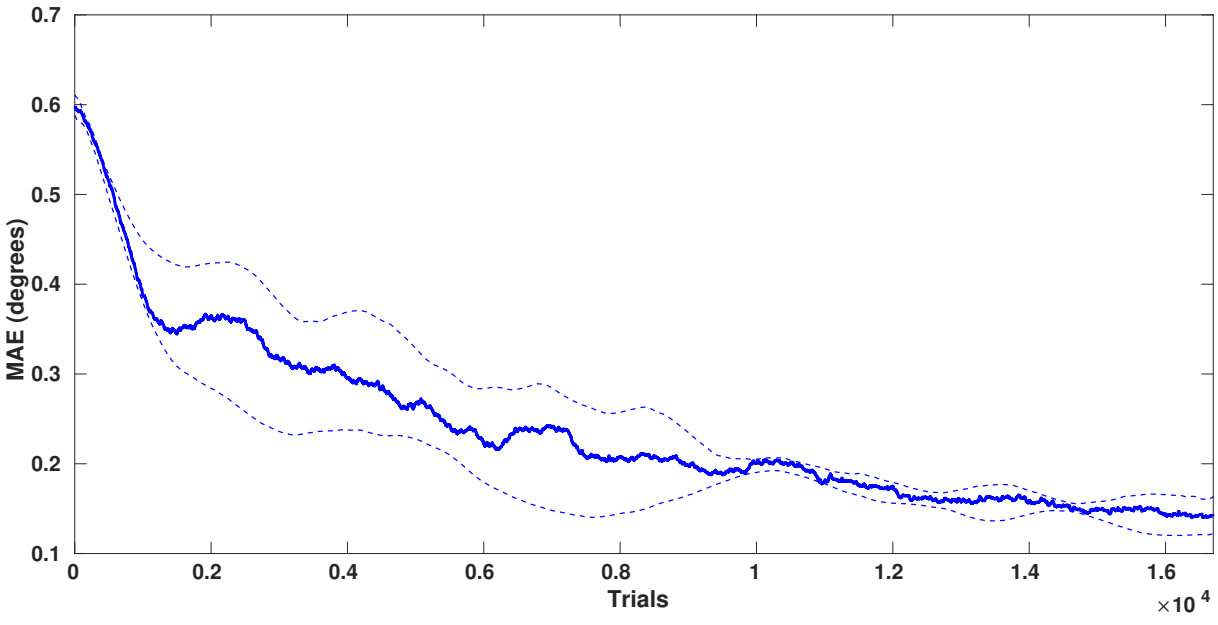


Figure 6.6: Eye movement MAE of single lateral position at 5 cm

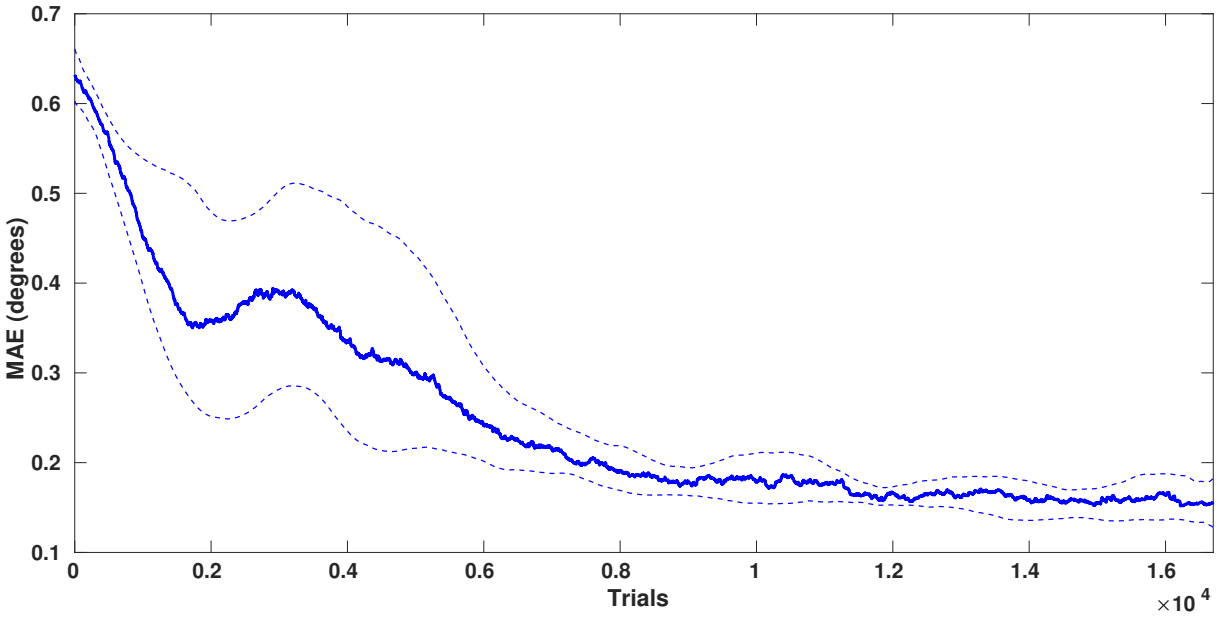


Figure 6.7: Eye movement MAE of single lateral position at 7 cm

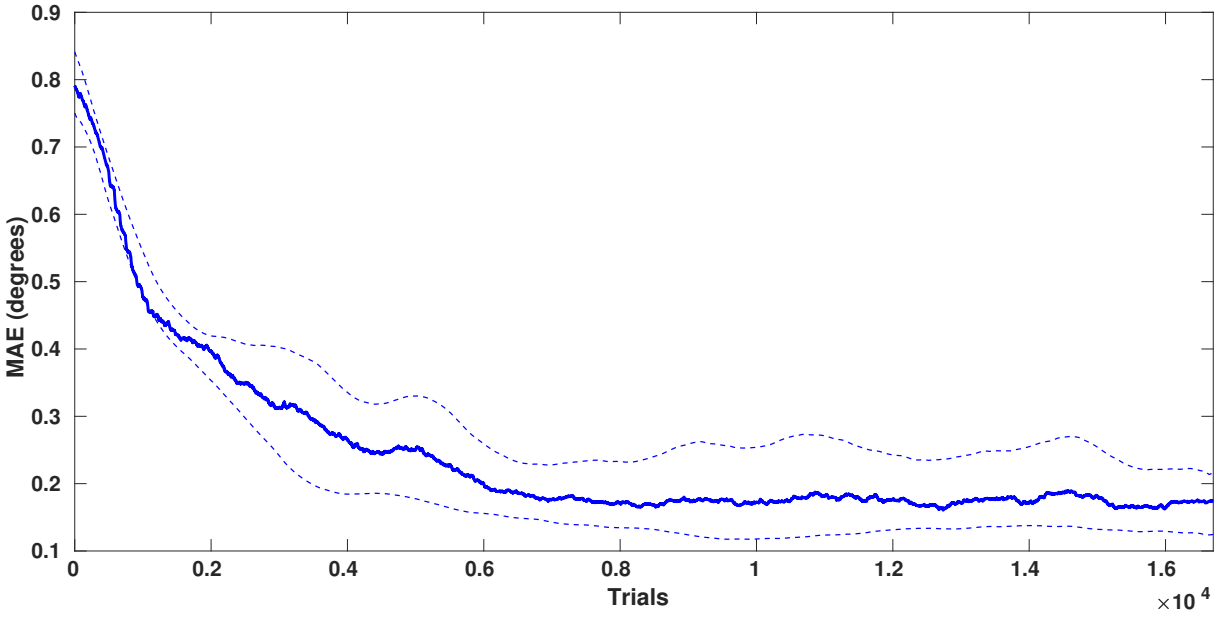


Figure 6.8: Eye movement MAE of single lateral position at 10 cm

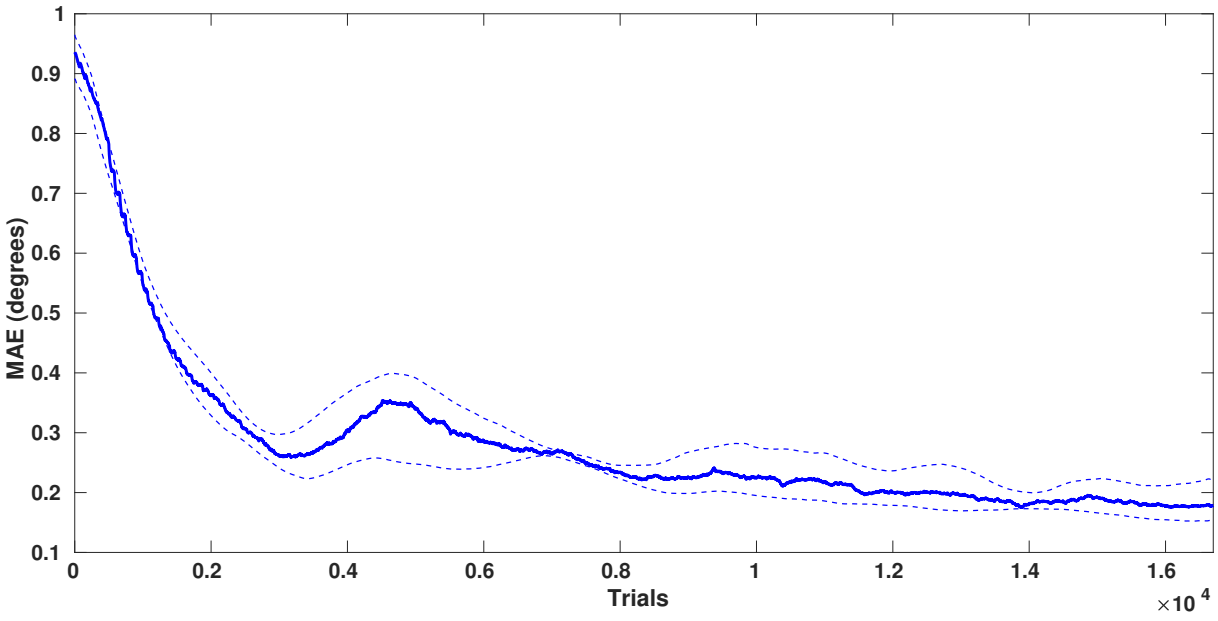


Figure 6.9: Eye movement MAE of single lateral position at 13 cm

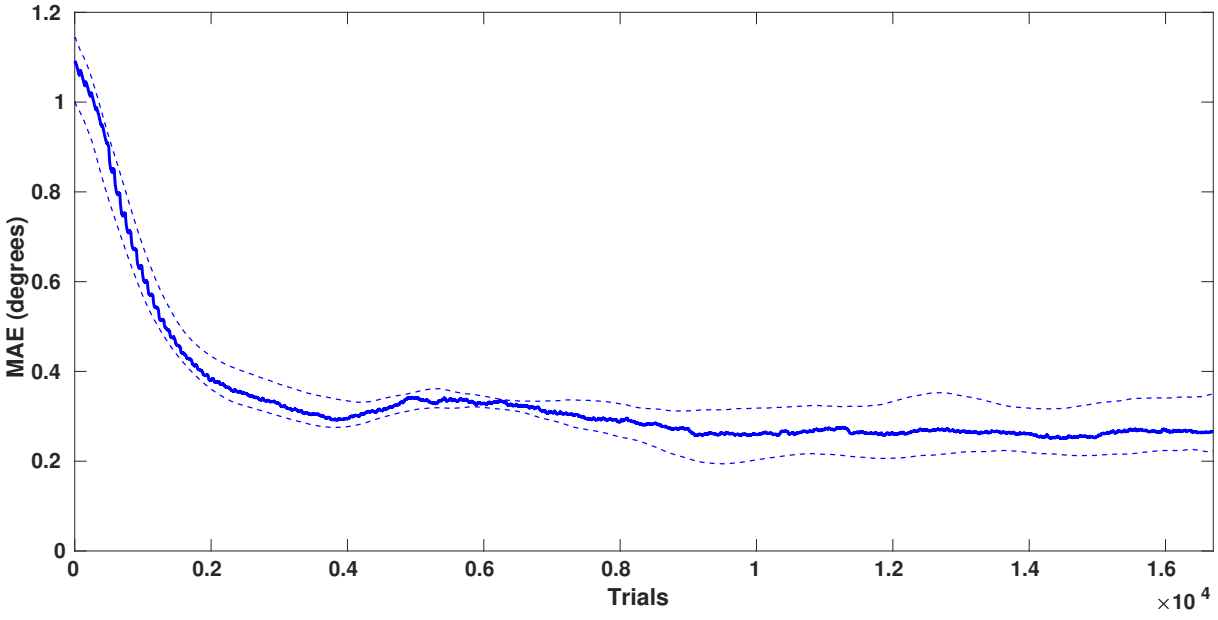


Figure 6.10: Eye movement MAE of single lateral position at 15 cm

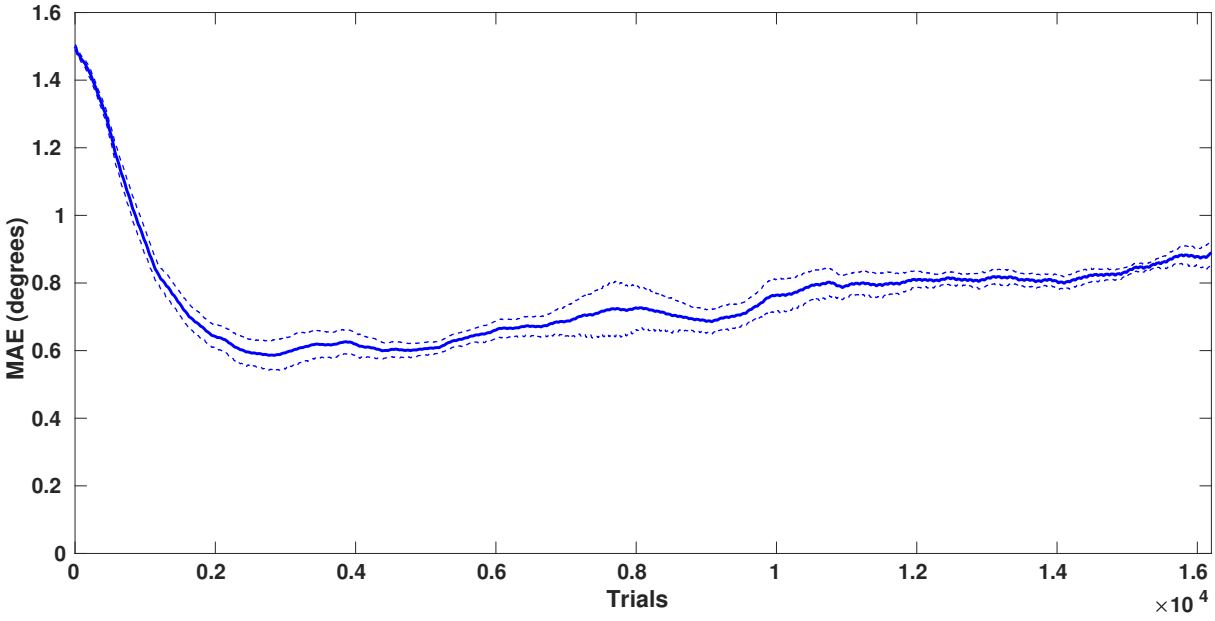


Figure 6.11: Eye movement MAE of single lateral position at 20 cm

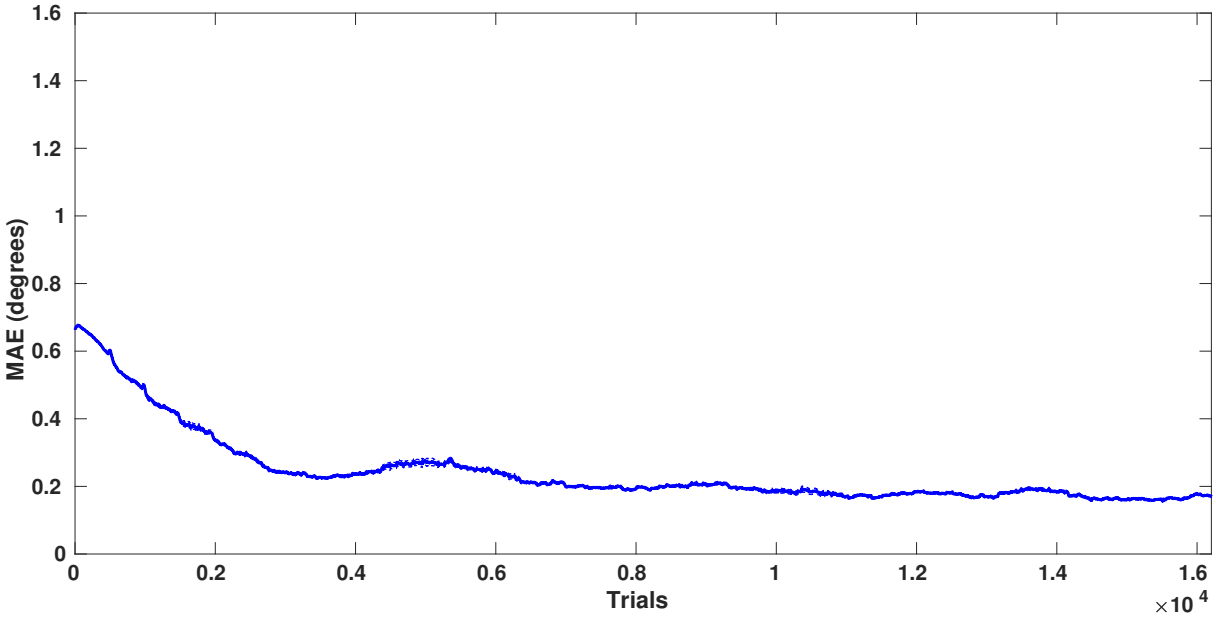


Figure 6.12: Eye movement MAE of multiple lateral position 5-10 cm

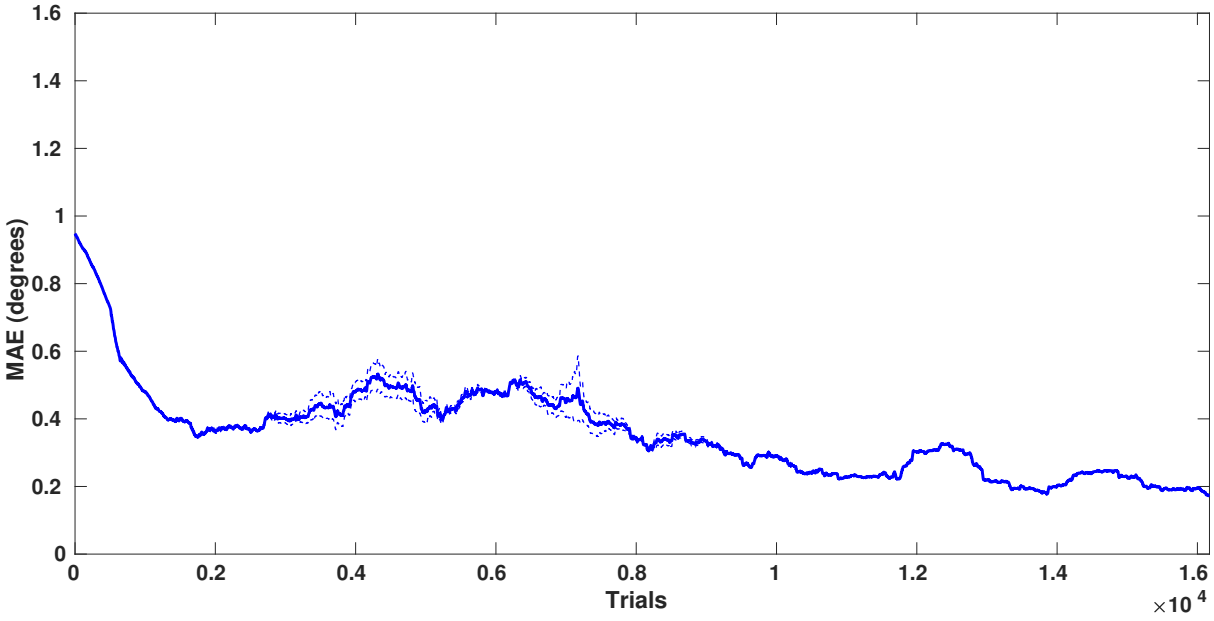


Figure 6.13: Eye movement MAE of multiple lateral positions 5-20 cm

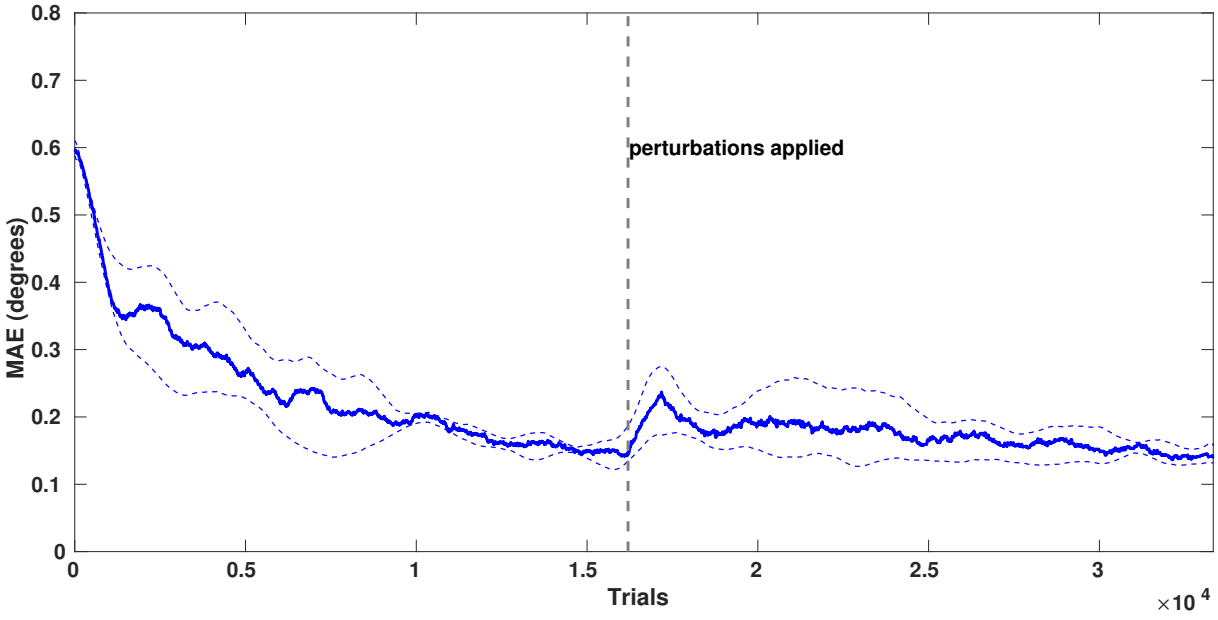


Figure 6.14: Eye movement MAE of single lateral position at 5 cm after the disturbances

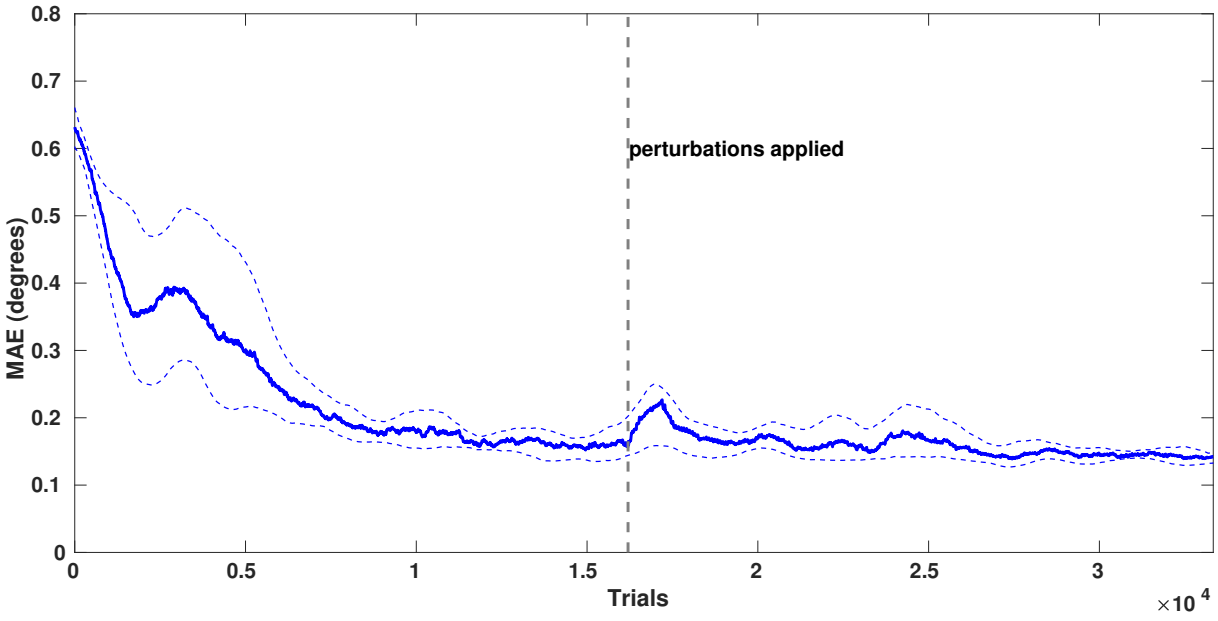


Figure 6.15: Eye movement MAE of single lateral position at 7 cm after the disturbances

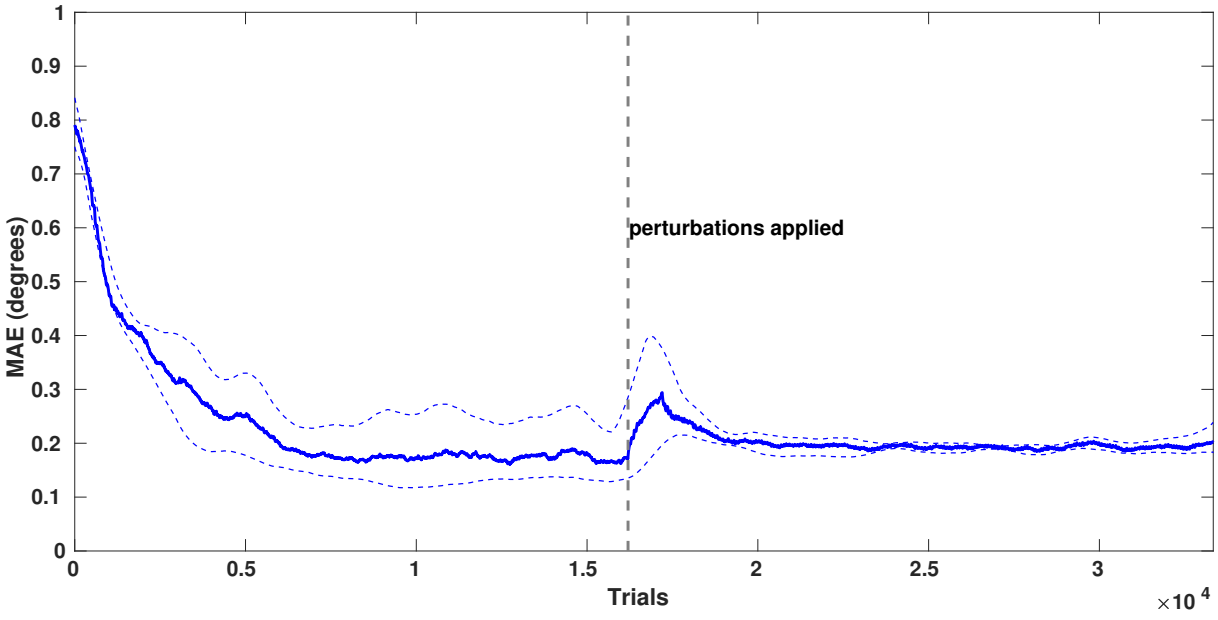


Figure 6.16: Eye movement MAE of single lateral position at 10 cm after the disturbances

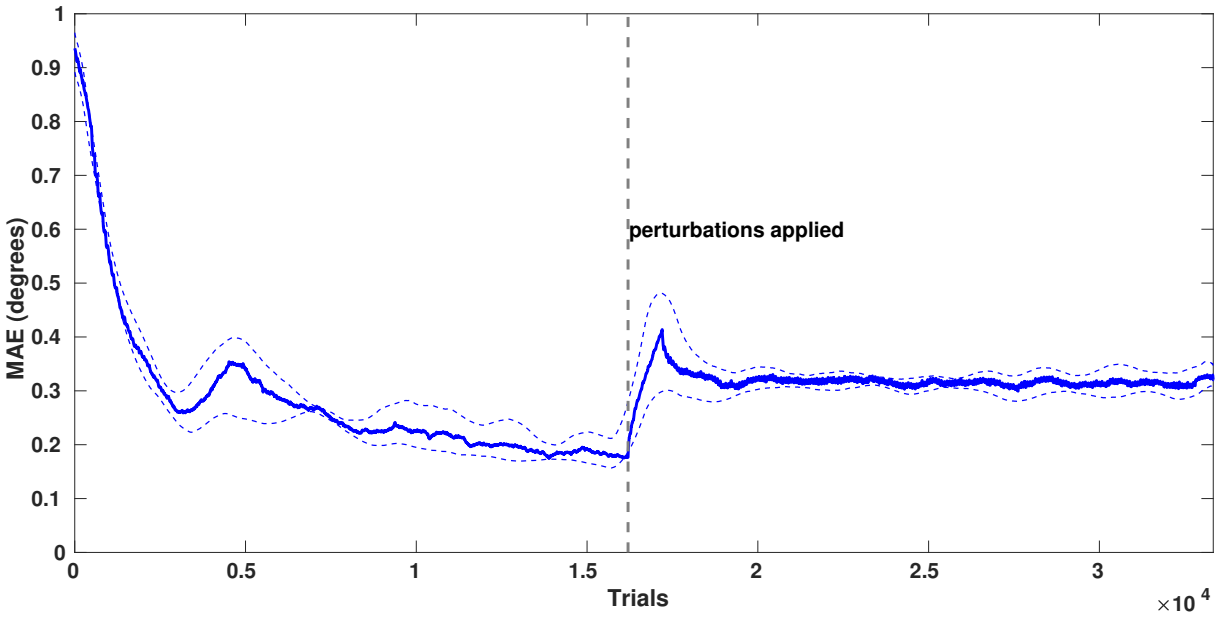


Figure 6.17: Eye movement MAE of single lateral position at 13 cm after the disturbances

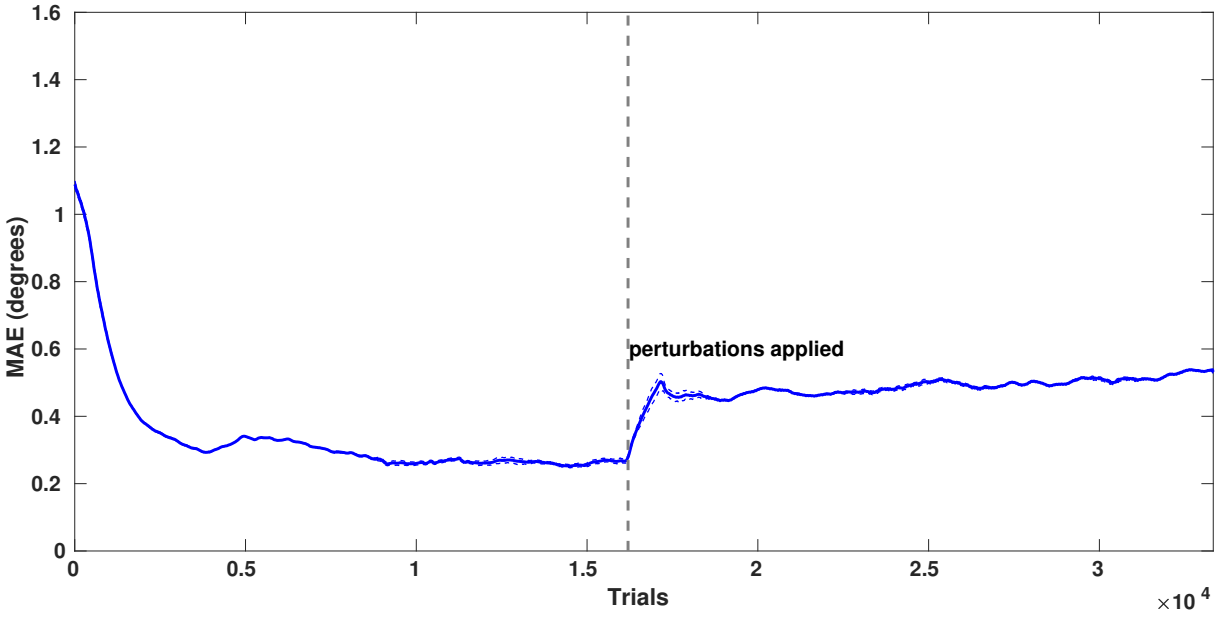


Figure 6.18: Eye movement MAE of single lateral position at 15 cm after the disturbances

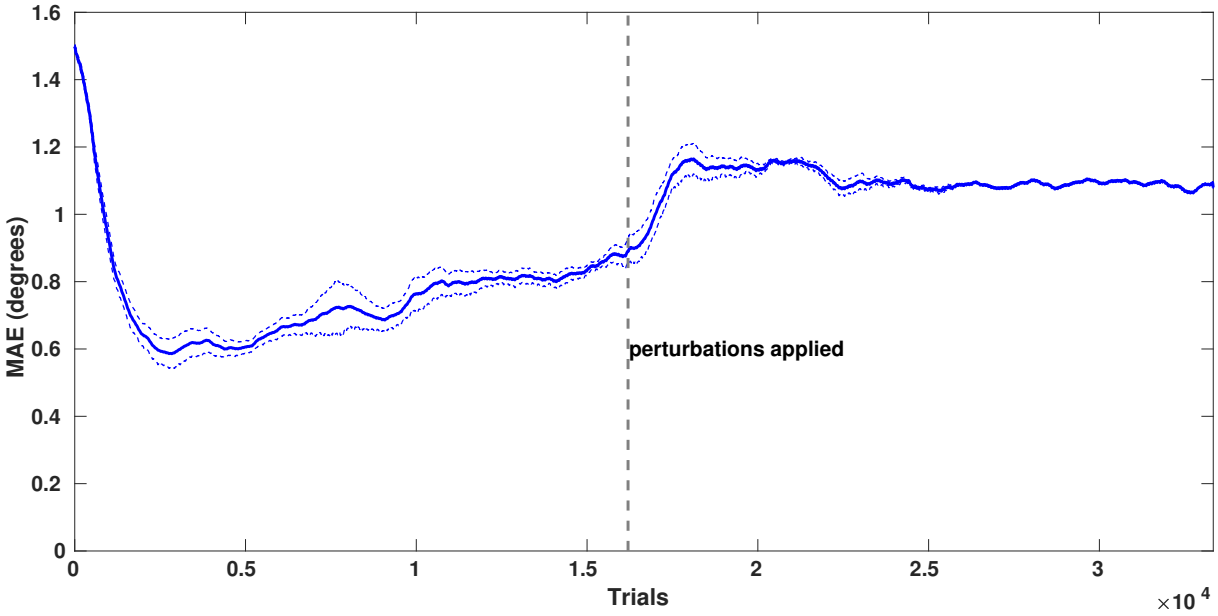


Figure 6.19: Eye movement MAE of single lateral position at 20 cm after the disturbances

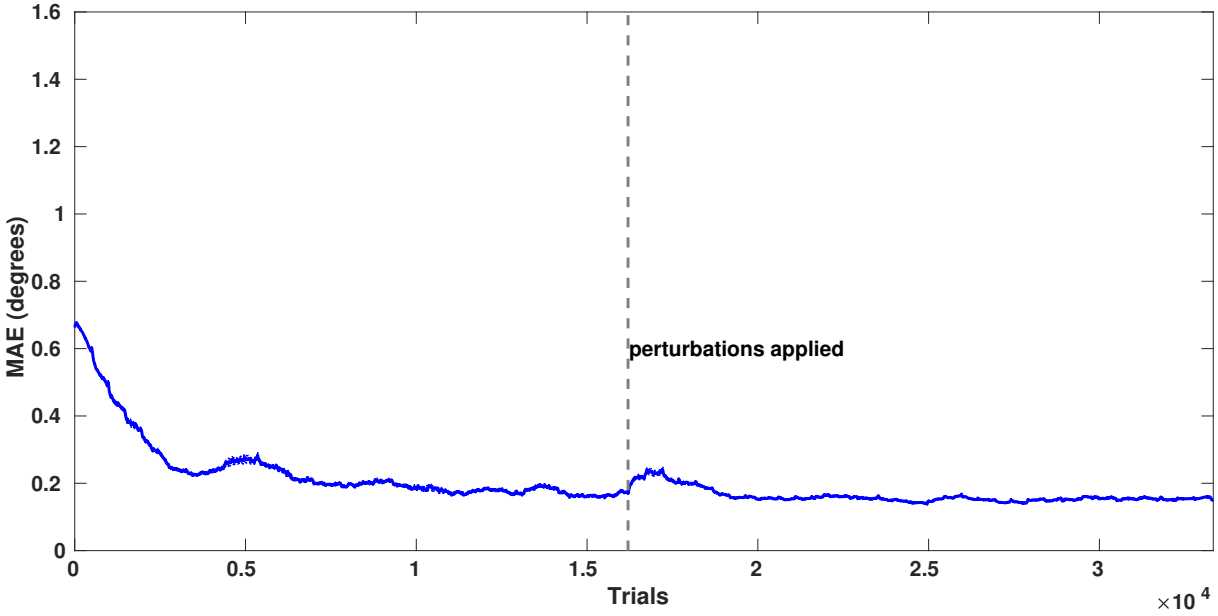


Figure 6.20: Eye movement MAE of multiple lateral positions 5-10 cm after the disturbances

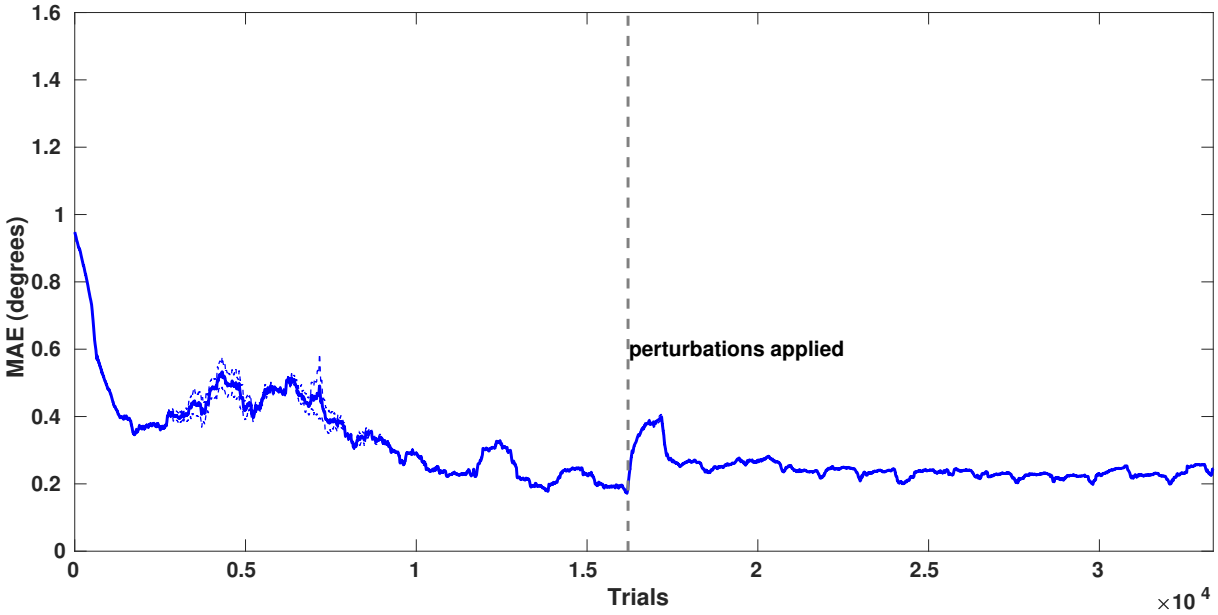


Figure 6.21: Eye movement MAE of multiple lateral positions 5-20 cm after the disturbances

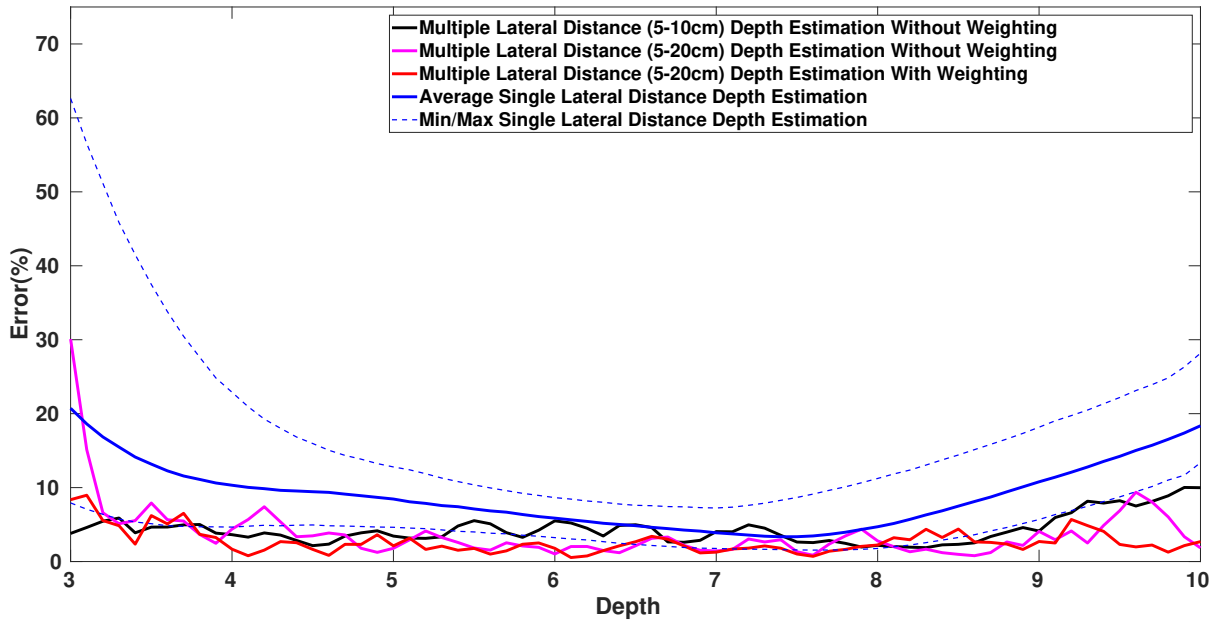


Figure 6.22: Distance estimation error

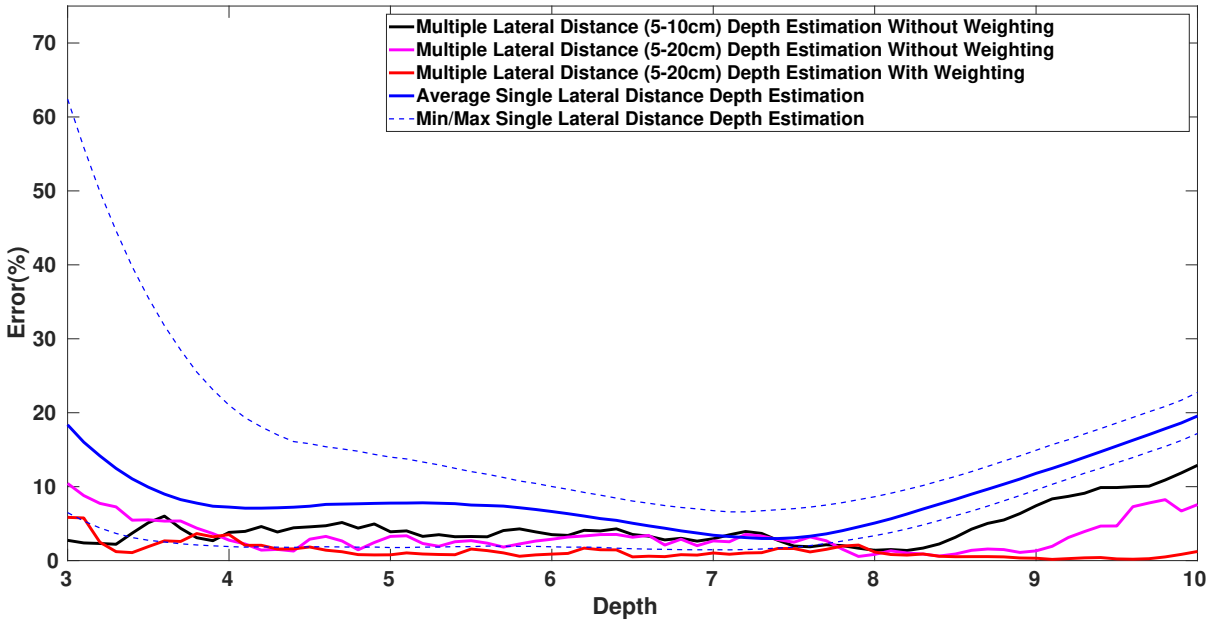


Figure 6.23: Distance estimation error at each distance after the disturbances

Chapter 7

Schemes of Motion Parallax Based with Optimal Lateral Movement

In the previous chapter, we propose a way to improve the framework by moving laterally at multiple position. However, in practice the robot movement should not be predefined. In this chapter, we will discuss how to make the robot learns to move optimally based on its training experiences.

7.1 Philosophy of This Work

To understand the surrounding environment, motion parallax phenomenon is one of the important key to be aware of depth in the environment. Motion parallax gives two kinds of depth information which are the distance from the observer to the fixating object d (egocentric), and the distance from the fixating object to another object M (allocentric), as shown in Fig. 7.1. Motion parallax involves 4 parameters which are the egocentric depth (d), allocentric depth (m), change in angle between the two objects (θ), and the angle required to fixate the object ($pan1$). By knowing the ratio of changes between the two angle and one kind of depth, we can determine another kind of depth [103, 104].

In the previous chapters, we successfully create the framework that utilizes motion parallax effect to estimate various depths. However, the lateral movement is predefined. The robot has to travel to the same lateral distance every time which is redundant for close visual stimuli. The study also does not cover allocentric depth which is meaningful

information provided from the motion parallax effect. To be able to recognize the relative depth, parallax angles may be observed during lateral body movements. Large lateral movement usually results in large parallax angle. Since a small parallax angle could be unnoticeable, the lateral movement should be large enough to provide a meaningful parallax information. In [105], they make experiments to understand how well human can distinguish between close and far objects. They found that there is a certain parallax threshold that effects the performance. So, a mechanism that selects optimal lateral body movement that is sufficient to differentiate allocentric depth can be based on the parallax threshold.

In this chapter, we propose a novel framework that lets robots understand the depth information provided by the motion parallax phenomenon while using an appropriate lateral movement. The framework considers three important mechanisms (1) visual sensory representation by sparse coding, (2) eye movement generation by reinforcement learning, and (3) understanding of allocentric depth information by utilizing the optimal lateral movement. This approach enables robots not only to autonomously learn sensory representation and eye movement controls but also to understand relative depth information from the motion parallax effect. The robot will be able to learn the optimal lateral body movement integrating with the developmental learning framework to differentiate two visual stimuli with allocentric depth.

7.2 Model Architecture

By letting the robot move laterally and capture the successive images, it can generate a motion parallax phenomenon under the different conditions, such as positions and speeds of its body. In this research, we assume that the robot can perfectly control their lateral body movements without uncertainty. The developments of related cognitive functions, such as visual representation and eye movements control, will only be focused on for understanding motion parallax and depth information in this study. Fig. 7.2 shows the architecture of the framework. The goal of this framework is to find lateral movement that generates parallax angle that is enough to distinguish visual stimuli which have variable textures at various depths and lateral positions. We utilize multiple sparse coding schemes as a sensory coding model coupled with reinforcement learners to achieve efficient coding

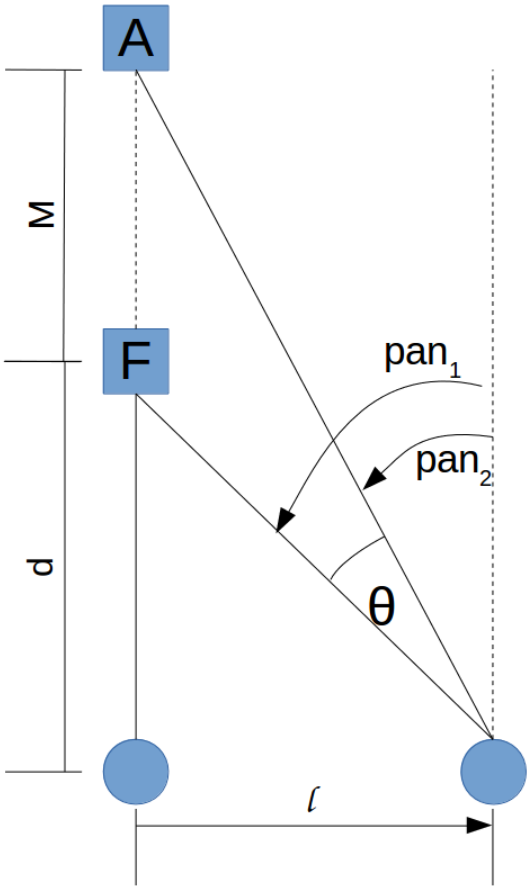


Figure 7.1: The model of parallax occurs when moving laterally by l. θ represents the parallax angle that is formed when focusing on fixating point F while there is another object A in the field of view. d is the egocentric depth between the robot and the fixating object. M is the relative depth between the two objects.

of the visual inputs from the camera. The sensory coding models learn to represent the input images, while the reinforcement learner learns to generate an action to increase the efficiency of the coding model.

7.2.1 Optimal Lateral Movement Selection

This section explains how the robot can select an optimal lateral movement based on the motion parallax effect.

Parallax Threshold

According to [105], there exists a certain parallax threshold for a human to be able to differentiate close and far objects. Depending on each person, the threshold that they can notice parallax are different. As shown in Fig. 7.1, the parallax, θ, is created after the observer moved laterally, while keep fixating at the object F. By using the fact that the observers can distinguish close and far objects if they could notice the change in parallax, it is possible to implement the optimal lateral movement selection. So, the parallax threshold is necessary to implement the optimal lateral selection for the robot.

Since the parallax threshold depends on how the observers could feel the movement of the images physically (eye's muscles) or visually (images on the retina), the parallax threshold for the robot can be assumed to be θ^* a multiple of smallest movements of the eye that the robot can make. For the relative depth, we assume that the robot is supposed to be able to classify objects that are placed M meters far from each other.

Comparing Parallax

We can find the parallax angles by taking subtraction of the two pan angles fixating on F and A, i.e. $pan_1 - pan_2 = \theta$. Then we can determine if the current lateral distance l is enough to distinguish two objects at depth-pair d and $f + M$ by testing if:

$$\theta \geq \theta^* \tag{7.1}$$

If the condition satisfies, it means that there is enough eye movement for the robot to detect between the two objects that are placed M far from each other. Figure 7.3 shows an example when $M = 0.2$ m and the objects are placed between 2 and 5 meters. The pan angles are calculated by hand using trigonometry. White cells are the position where the condition is met and vice versa for the black cells. It represents distinguish-ability of the two objects placed in the different depth (depth-pair) with respect to the lateral distance l. We can see that large lateral movement can mostly differentiate most of the depth-pair, while the small lateral movement can barely see the difference between depth 2.0 and 2.2 meters.

The Selection

Optimal lateral movement selection is a mechanism that helps the robot decide at which lateral position is enough to differentiate objects at a different depth. As shown in the previous section, we may observe the parallax angles and then compare them with the parallax threshold to know if the lateral movement was enough for the robot to distinguish the depth-pair or not. Then, we can determine lateral movements that are effective for each depths. However, in the robot application, the distance between the robot and the objects is unknown. So, it is not possible to calculate the optimal lateral movement. In that way, the robot has to learn how to generate optimal lateral movements.

In this research, the robot moves laterally step by step. At each step the robot moves by l_s meters, while it tries to sequentially fixate two visual stimuli which are placed M meters far from each other with two pan angles, $\alpha_{j,i}$ and $\alpha_{j,i+1}$. The robot moves until it reaches the optimal lateral distance l^* or the maximum lateral distance $l^* = l_m$. The robot then moves back to the center (zero lateral distance). Every time the robot return to the center, the robot moves backward for another M meters to increase the distance between itself and the visual stimuli. The robot moves backward until it reaches the maximum depth $d = D$, which then it will return to the original position. This is considered as one-trip.

To enable the robot to learn optimal lateral movement, we assume that the robot can remember eye movements at each lateral position for all q depths after one trip which can be represented by:

$$X = \begin{bmatrix} \vec{\alpha_1} & \vec{\alpha_2} & \cdots & \vec{\alpha_q} \end{bmatrix} \tag{7.2}$$

$\vec{\alpha_i} = \begin{bmatrix} \alpha_{j,i} & \alpha_{j+1,i} & \cdots \alpha_{r,i} \end{bmatrix}^\top$ is a column vector containing $r = \frac{l_m}{l_s}$ pan angles required to fixate i-th depth from all of the lateral movements. Then the parallaxes can be calculated by subtracting each column with its next column as follows:

$$Y = \begin{bmatrix} \vec{\theta_1} & \vec{\theta_2} & \cdots & \vec{\theta_{q-1}} \end{bmatrix} \tag{7.3}$$

where, $\vec{\theta_i} = \vec{\alpha_i} - \vec{\alpha_{i+1}}$. Since the robot could make mistake on generating eye movement which directly effects the parallax angles. To prevent the problem, an additional condition is presented. Since the robot always moves far away from the objects, we can safely assume

that the lateral movement should be an increasing function, i.e. the lateral movement in the farther depth should be larger than the close depth. Thus, the new condition is:

$$l(t) \geq l_p^* \tag{7.4}$$

, where l_p^* is the optimal lateral movement from the previous depth (the lateral position that the robot stopped).

Every time the robot moves laterally, it checks $\theta_{j,i}$ in Y if it satisfies the Eqn. 7.1 and Eqn. 7.4. If the condition was satisfied, then the robot consider the current movement as the optimal lateral movement. The robot then proceeds to move backward and continue the procedure. Otherwise, the robot continues moving laterally.

7.2.2 Sensory Coding Model

At the center position (zero lateral distance), an image I_0 is captured from the camera as a reference. The robot then moves laterally by one step $l(t) = l(t-1)+l_s$. An image $I_p(t)$ is captured from the camera, $p = \frac{l(t)}{l_s}$. The robot then generates a smooth pursuit eye movement by using two captured images $I(t) = \begin{bmatrix} I_0 & I_p(t) \end{bmatrix}$ to learn sensory representation of motion parallax and smooth pursuit eye movement for the camera.

After generating smooth pursuit eye movement for h iterations, i.e. one trial, the robot continues to move laterally to the next position. This process is repeated $p_m = \frac{l^*}{l_s}$ times reaching the optimal lateral position l^*, changing the depth and the texture of the visual stimulus.

The two input images in the matrix $I(t)$ are cropped by 250x250 pixels and 150x150 pixels from the center of the images which will be sub-sampled. The two cropped images represent a fine scale and a coarse scale, respectively. The two scales of the images represent the foveal system in human eyes. The fine scale image represents a foveal region in our eyes which can catch more details at the center of vision. While the coarse scale represents a parafoveal area which contains smaller details. Discussions and comparisons of using multiple scales of images have been done in [48]. They discussed how gaining the access to multi-scale images could improve the learning of the framework. On the other hand, having only one scale might prevent the system from learning appropriately.

After the cropping, the cropped images are converted to grayscale. 10 by 10 pixels

patches are then extracted from the grayscale images, whose locations are generated by 1 pixel and 4 pixels shifts horizontally and vertically for coarse scale and fine scale, respectively. The image patches are then sub-sampled by using the Gaussian pyramid algorithm which has a sub-sample factor of 8 for the coarse scale, and 2 for the fine scale. The patches are reshaped and normalized to be one-dimensional vectors which have zero mean and unit norm, $\gamma_i^j(t)$. i is the index of the patch, which $j \in \{C, F\}$. C is for the coarse scale, and F is for the fine scale. With the sub-sampled images, the framework will be able to handle image disparity that is larger than patch width. Note that the fine-scale helps in fine-tuning the eye movements.

The two one-dimensional vectors are then combined into a single vector $\gamma^j(t)$. The first 100 elements of the vectors are from the first image I_0 and the remaining are from the second image $I_p(t)$. The result vectors ($\gamma^C(t)$ and $\gamma^F(t)$) consist $K = 200$ elements.

Initially, two dictionaries, $\phi^j(t) = \{\phi_n^j(t)\}_{n=1}^N$, are randomly generated using a uniform distribution. Each of the dictionaries contains $N = 288$ basis functions $\phi_n^j(t)$. One dictionary is for the coarse scale, and another one is for the fine scale. Later, the patches are encoded by the sparse coding algorithm in a linear fashion. Each patch can be represented by a linear combination of the basis functions picked from the coarse or fine scale dictionary.

We use the matching pursuit algorithm [106] to estimate and find the sparse representation of the input vector by the weighted sum as follows:

$$\gamma_i^j(t) \approx \hat{\gamma}_i^j(t) = \sum_{n=1}^{N} b_{i,n}^j(t)\phi_n^j(t) \tag{7.5}$$

The matching pursuit algorithm suits to concept of sparse coding, which can estimate $\Gamma_i(t)$ by using a limited number of coefficients. In this research, the maximum number of non-zero scalar coefficients $b_{i,n}(t)$ is set to be 10 elements to ensure sparseness of the efficient coding. For later use in reinforcement learner part, pooled activity, $f_n(t)$, which represent the activity of each neuron cell is calculated from the coefficients from matching pursuit algorithm as follows:

$$f^j(t) = \begin{bmatrix} f_1^j(t) \\ f_2^j(t) \\ \vdots \\ f_K^j(t) \end{bmatrix} . \tag{7.6}$$

Where, each element of the vector $f^j(t)$ is described as:

$$f_n^j(t) = \sum_{i=1}^{K} b_{i,n}^j(t)^2. \tag{7.7}$$

A reconstruction error is introduced as a cost function to be used in the sensory coding model and the reinforcement learner. It measures the estimation error of vector $x(t)$. The reconstruction error is defined as:

$$e^j(t) = \frac{1}{K} \sum_{i=1}^{K} \frac{\|\gamma_i^j(t) - \sum_{n=1}^{N} b_{i,n}^j(t)\phi_n^j(t)\|^2}{\|\gamma_i^j(t)^2\|}. \tag{7.8}$$

Gradient descent method is used to update the dictionaries with the reconstruction error as the cost function. After each updates, the dictionaries are normalized.

7.2.3 Reinforcement Learning

The state representation of the reinforcement learner can be directly described by a combination of the coarse scale and the fine scale pooled activity, $f_n(t)$ as follows:

$$f(t) = \begin{bmatrix} f^C(t) \\ f^F(t) \end{bmatrix} . \tag{7.9}$$

The reward that is given to the learning agent is a negative summation of the reconstruction error from both of the image scales which is described as:

$$R(t) = -(e^C(t) + e^F(t)) . \tag{7.10}$$

The actor-critic algorithm number 3 proposed in [95] is employed for the leaner agent. For action selection, we use Gibbs distribution (softmax) for probabilistically choosing an action as follows:

$$\pi(f(t), a_t) = \frac{e^{z_a}}{\sum_{a' \in A} e^{z_{a'}}} . \tag{7.11}$$

For each action, the activation value z_a is given by:

$$z_a = \sum_{n=1}^{N} w_a(t) f_n(t) , \tag{7.12}$$

where $w_a(t)$ is a weight vector from the state $f(t)$ to action a. The action is a pan angle of the cameras in degrees. Possible actions a are contained in a set of actions A. In this research we use $A = \{-0.2°, -0.1°, -0.05°, 0°, 0.05°, 0.1°, 0.2°\}$. Thus, the policy maps $f(t)$ to $a \in A$.

7.3 Simulations & Results

In this section, we evaluate the framework by observing and analyzing the lateral movement together with the eye movement generation.

7.3.1 Experimental Setup

We use V-REP, a robot simulator, as a 3D environment visualization for the framework and the framework is implemented on MATLAB. The environment in the simulator comprises HOAP3 robot model, an object with interchangeable textures, and a still background image as shown in the top picture in Fig 7.2. We assume the parallax threshold θ^* for the robot to be 2 times of the smallest eye movement in A which is 0.1°. In this experiment, we virtually simulate the two visual stimuli by making the robot moves back for another $M = 0.2$ m instead, while the texture is also changed. To capture the successive image frames, the lateral movement of the robot is simplified to be pick-and-place and the amounts of movement with $l_s = 0.01, l_m = 0.15$. The maximum depth is set to be $D = 4.8$ m (the distance between the robot and the farthest stimuli is 5 m). 100 images are prepared as the texture of the visual stimuli. Each image is iteratively trained with $h = 30$ iterations. We conduct 4 experiments with the same setup.

7.3.2 Eye Movement Analysis

In this section, we analyze the eye movements performance of the framework. Eye movement generation is observed to verify the progress of learning. Eye movements at the end of each trial are recorded and compared with the desired eye movement. The mean absolute error (MAE) is computed to evaluate the training as shown in Fig. 7.4. The black dashed line represents the variance between the simulations. We can see that the robot learns and improve eye movement generation over time. Since the visual stimuli are changed periodically, the increased in MAE is presented. Because the robot has to learn to create the new visual representation of the new textures. The 4 simulations do not differ from each other much as seen with the variance.

7.3.3 Optimal Lateral Movement

In this section, we visualize and validate the robot's choice of the lateral movement. Figure 7.5 shows a heat-map of the chosen lateral movement averaging from the 4 simulations in each trip starting from the top row. The amount of the lateral body movement is represented by the color bar. The bottom row shows the expected lateral movement that is calculated by hand. We can see that most of the trips, the robot chose the amount of the lateral movement near the expected lateral movement in the bottom row.

To further understand how the chosen lateral movement affects the distinguish-ability of the depth-pair, we may see Fig. 7.6. The figure tells how well the robot can distinguish the depth-pair. The color's value represents the number of simulations that could successfully differentiate the depth-pair. e.g. White means 4 simulations can distinguish the depth-pair. Also, similar to the previous figure, the bottom row represent the expected classification. The figure shows that the robot can classify most of the close depth depth-pair (left side). However, at the far depth (right side) the robot could not distinguish the depth-pair most of the time, which follows the expectation as seen in the bottom row. However, some cells in the right side report as a success (gray-white) which does not follow the expected value. The reason is simply that there still exist eye movement errors as shown in the previous section.

To summarize the results from the experiments, Fig. 7.7 shows the distinguish-ability map for the 4 simulations similar to the Fig. 7.3. We can directly compare them together since Fig.7.3 is made up with the same set-up. Similar to the previous figure, the color bar's value represent the number of simulations that can distinguish the depth-pair. We can see that the chosen lateral movement are similar to the example shown in Fig. 7.3. This tells us that it is possible for the robot to learn how to generate optimal lateral movement. In addition, averaging all of the 4 simulations, the robot saves 25.22% of the lateral movement.

7.4 Summary

In this chapter, we propose the framework with the learning strategy that enables the robot to learn optimal lateral movement to distinguish two depths. While it can actively

generate an accurate smooth pursuit eye movement for various ranges of motion parallax during self-induced lateral body movement. The proposed framework can simultaneously learn to choose eye movements, select the optimal lateral movement, and create visual representations to understand the motion parallax effect. This research has proven that it is possible for the robot to find suitable lateral movements that are enough to differentiate allocentric depth of two visual stimuli that are closed together. In addition, the framework still remains the extendability of the egocentric depth estimation.

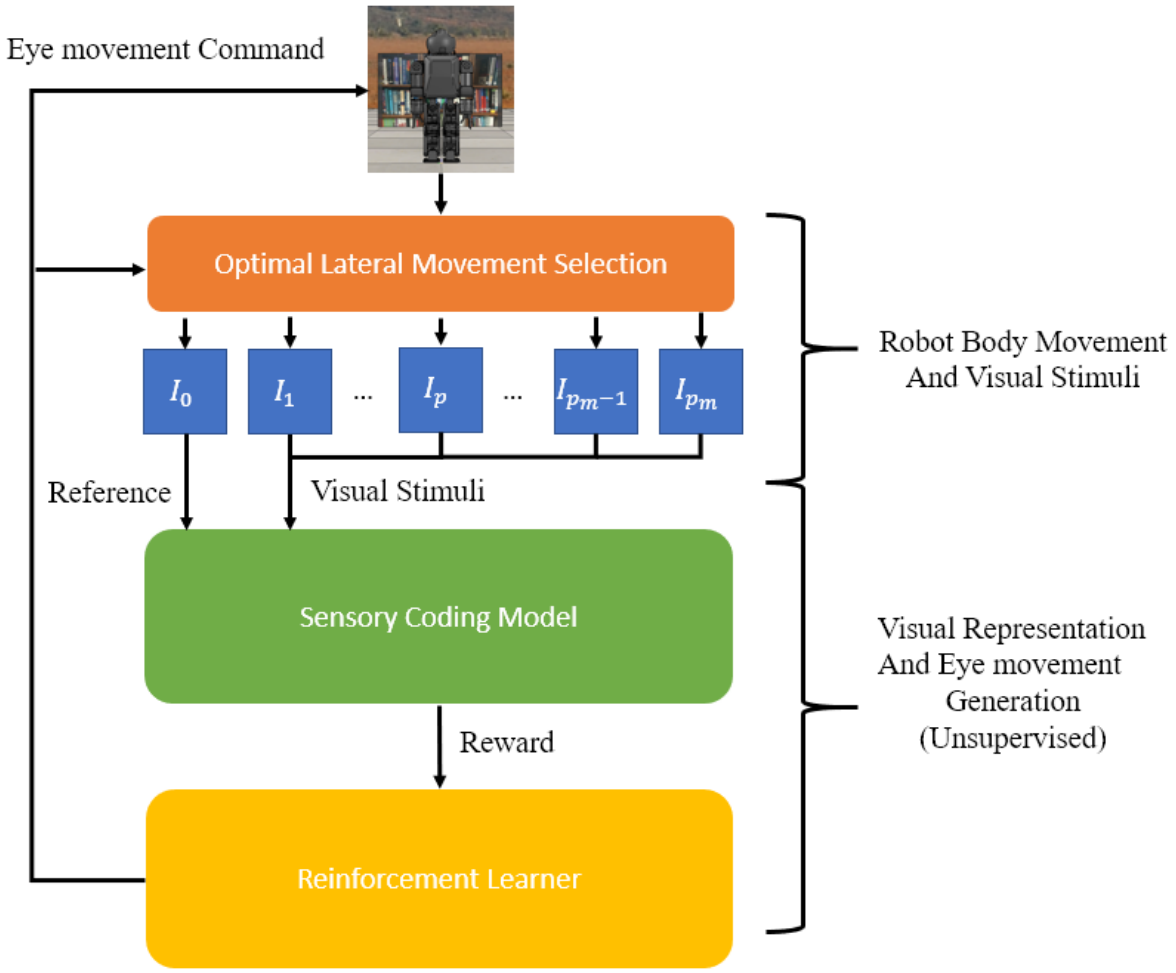


Figure 7.2: The robot then captures a reference image and then moves to the lateral position $l(t)$. To perform the motion parallax, the successive images I_0 and I_p are input into the sensory encoders with multiple image scales. Then, an output reward signal generated from the sensory encoders is sent to the reinforcement learner to generate an appropriate eye movement to hold the fixation during the body movement. Finally, a pan command is sent to the robot and it generates the smooth pursuit eye movement to maximize the redundancy between the successive images.

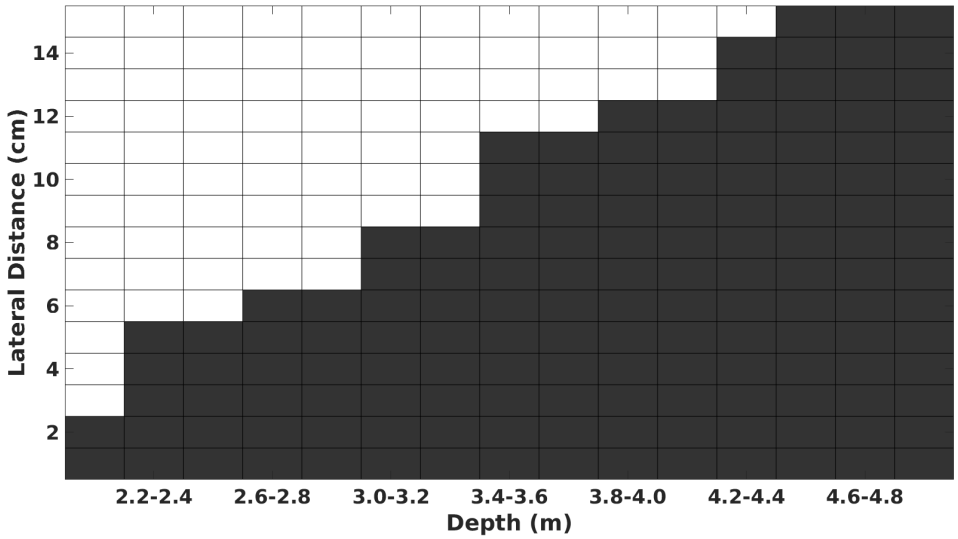


Figure 7.3: Distinguish-ability between each pair of depth (depth-pair). Black represents ambiguous depths that are difficult to distinguish with respect to the lateral distance. While white shows the depths that are easy to distinguish. e.g. at lateral distance 10 cm, it can easily tell the difference between depth 3.3 m and 3.4 m and the earlier depth-pairs. However, it can't distinguish the depth-pairs from 3.4 m

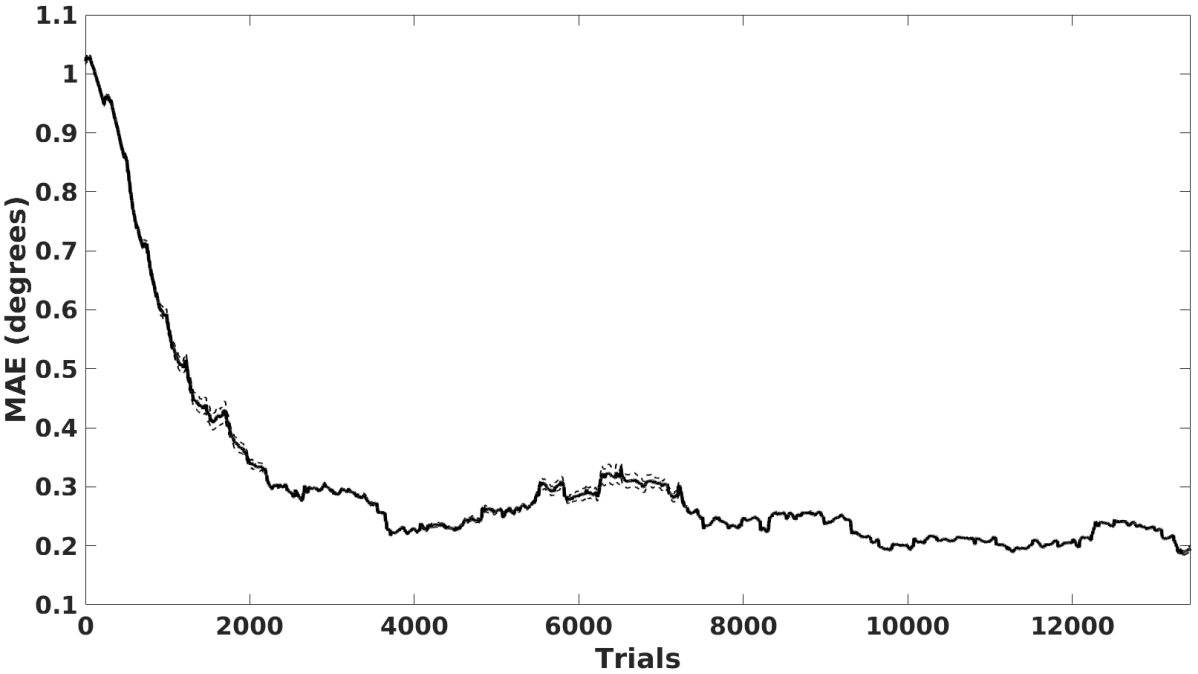


Figure 7.4: Eye movement mean absolute error (MAE) of the 4 simulations. Dashed lines represent the variance between the simulations

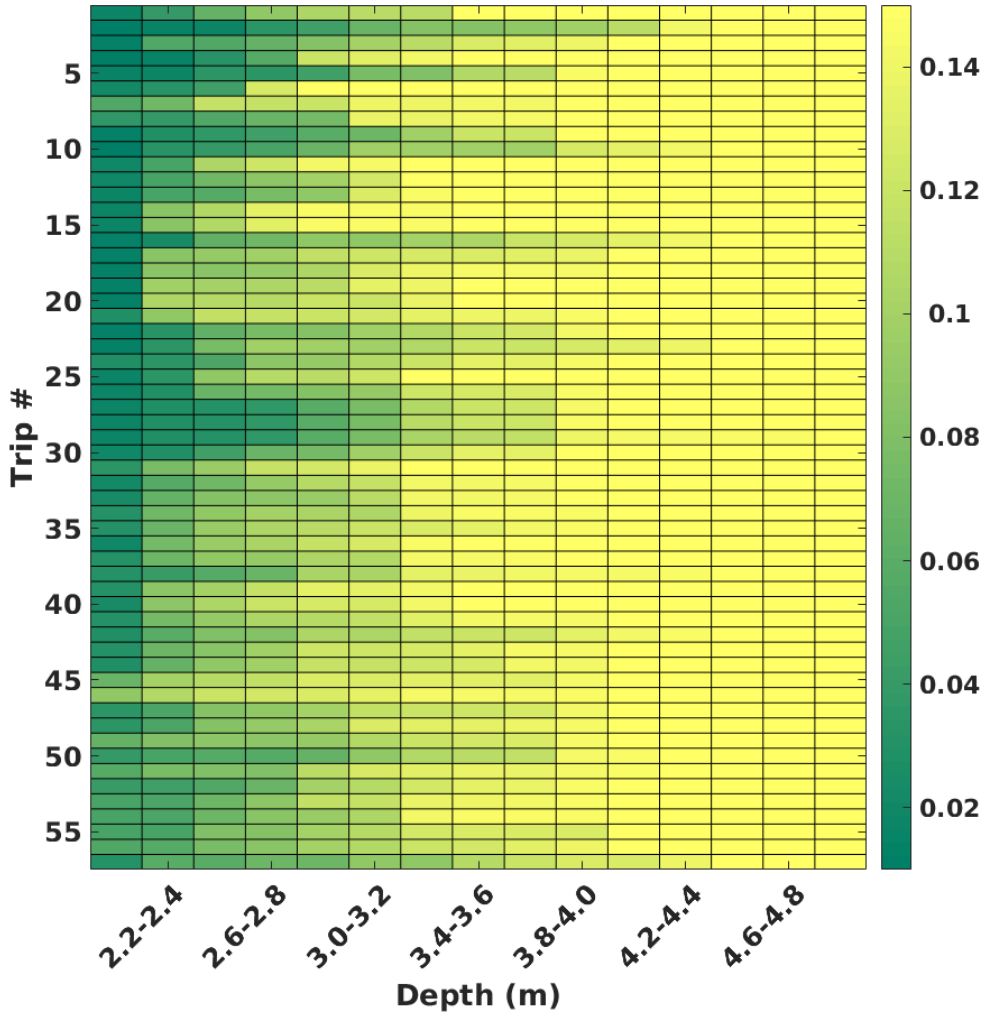


Figure 7.5: Heat-map represents the chosen lateral movement. Each row represents lateral movement that the robot chose to stop at each trip. The bottom row shows the expected lateral movement

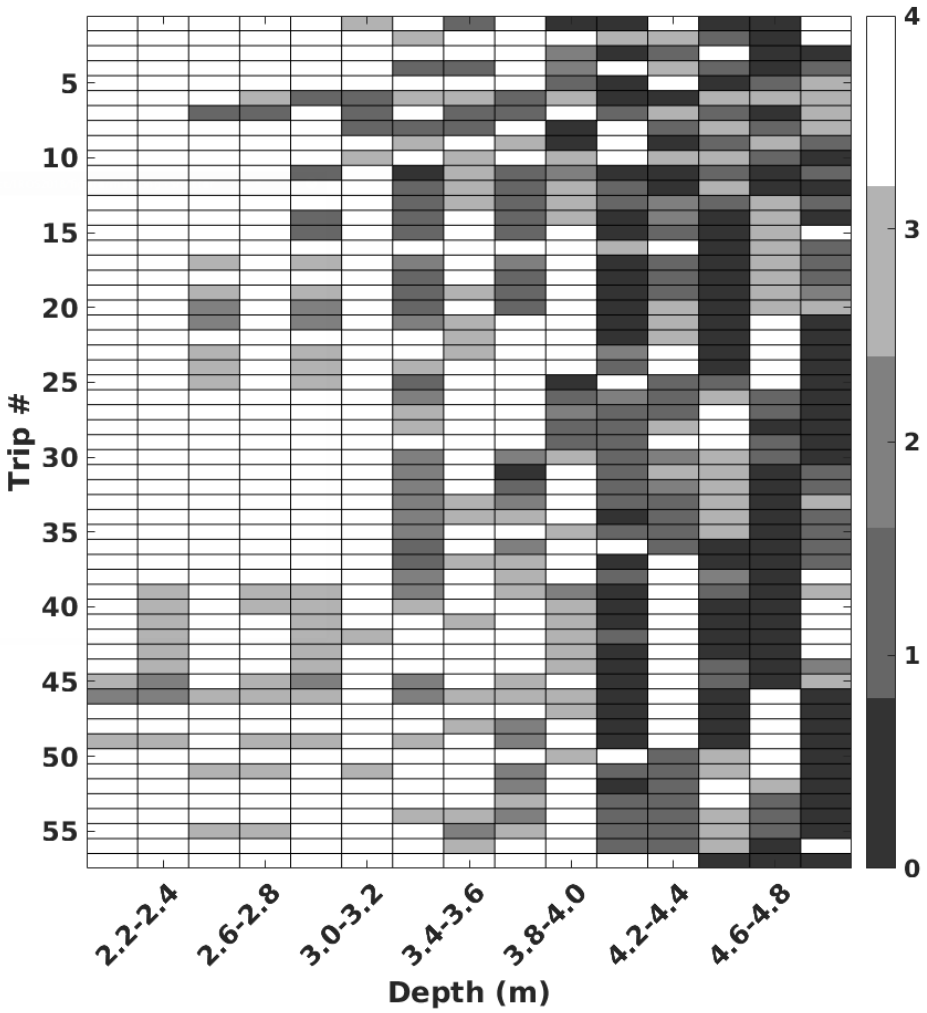


Figure 7.6: Classification for each depth-pair from each trip. White(4) means all of the 4 simulations can successfully differentiate the depth-pair, while black(0) means none can distinguish the depth-pair. The bottom row shows the expected classification

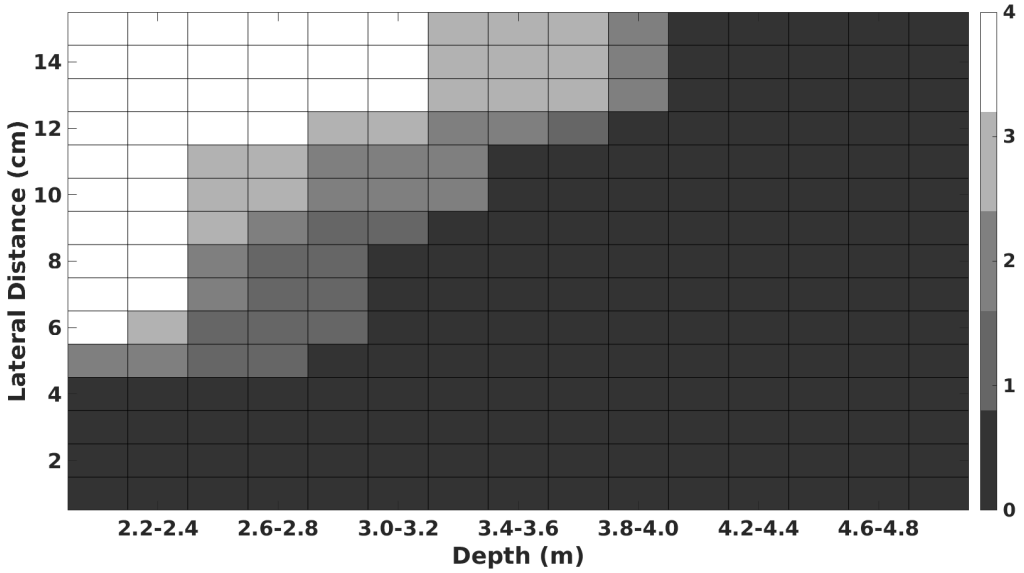


Figure 7.7: Distinguish-ability between each pair of the depth of the 4 simulations from the final trip. Black represents ambiguous depths that are difficult to distinguish. White and gray represent how many simulations can distinguish the depth-pair. e.g. at lateral distance 10 cm, there are 3 simulations that can differentiate depth 2.4 and 2.6.

Chapter 8

Integration of the Motion Parallax and Stereo Vision

In this chapter, we propose a new strategy to combine the frameworks from the chapter 4 and 5 into a unified framework. The framework will be able to learn to generate both of the vergence and smooth pursuit eye movement.

8.1 Philosophy of This Work

In the previous chapters, we considered only the smooth pursuit eye movement and the vergence eye movement in isolation. Interestingly, developed organisms do not use only one visual depth cue for their whole lifetime. They can integrate the information about multiple visual depth cues and analyze the eye movements to perceive the spatial information about the surrounding environment.

In [49], they have successfully demonstrated generating multiple eye movements, which are smooth pursuit and vergence to track a moving object, but depth perception is not included in the learning framework. Moreover, all of the generated eye movement information could not be used for depth perception because stationary observer cannot extract depth information from motion parallax or optic flow without a priori knowledge such as object size.

Generally, in psychology, dominant eye is a concept that implies that one eye moves before another eye does. Recently there are studies that support the dominant eye hypoth-

esis [107–109]. Also, according to [110], they reported that when a motion is self-induced by active observer, two visual depth cues (stereo vision and motion parallax) will be sequentially activated which is not observable in a static observer. Therefore, we may consider that two eye movements for different visual depth cues during the self-induced lateral body movement can be sequentially generated in an independent process to minimize the conflict of multiple cues and then finally multiple eye movements are used to analyze the depth information by integrating each of them. This approach enables to autonomously learn not only sensory representation and eye movement controls for the multiple visual depth cue analysis but also active depth perception during self-induced body movements.

8.2 Model Architectures

In this chapter, we present a dominant eye strategy to combine two cues which are stereo vision and motion parallax together. One of each eye is assigned to be dominant and non-dominant eye. Dominant eye is responsible for motion parallax effect, while the non-dominant eye is for stereo vision.

The framework (Fig. 8.1) is divided into 2 parts which are motion parallax and stereo vision. The robot observe motion parallax effect through the dominant eye first, then followed by the stereo vision with the non-dominant eye, i.e. the two cues are performed sequentially. One iteration t is divided into 2 steps k_1 and k_2. First, at k_1 the robot moves laterally from the original position to the leftmost position to observe the motion parallax and generating the smooth pursuit eye movement. Then, at k_2 the robot perform the stereo vision just after the smooth pursuit eye movement is done by using vergence eye movement. After h iterations, the robot moves to the rightmost position. Then the robot observe the motion parallax and stereo vision respectively for another h iterations. Finally, the texture of the object and the depth between the robot and the object are changed.

Step k_1: images $I_{m,k_1}(t) = \begin{bmatrix} I_{m_1}(t) & I_{m_2}(t) \end{bmatrix}$ from the dominant eye are input to the framework to learn the sensory representation and smooth pursuit eye movement. The images are captured at different position before and after the lateral body movement.

Step k_2: after k_1, images from both dominant eye and non-dominant eye are captured,

$I_{s,k_2}(t) = \begin{bmatrix} I_{s_1}(t) & I_{s_2}(t) \end{bmatrix}$. The stereo images are then input to the framework to learn the sensory representation of stereo disparity and vergence eye movement.

8.2.1 Sensory Coding Model

The images that are input in the sensory coding model are treated as follows. Two input images are cropped by 128x128 pixels and 80x80 pixels at the center. The cropped images are converted to gray scale. Then they are extracted to multiple of 10x10 pixels patches which the locations are generated by 1 pixel and 4 pixels shift horizontally and vertically for the 128x128 window and 80x80 window respectively. The image patches are then sub-sampled using Gaussian pyramid algorithm by a factor of 8 for the larger window and factor of 2 for the smaller window. The two cropping sizes represent the foveal system in the human eyes. The smaller windows represents the foveal region (fine scale) in our eyes which can capture more detail more than the parafoveal area (coarse scale) which is represented by the larger window.

The image patches are reshaped to one-dimensional vectors which have zero mean and unit norm. The resulted vectors from the first image is then concatenated with the one from the second image resulting in a single vector, $x_i^j(t)$ consisting of $P = 200$ elements. Where, i is the index of the patch, and $j \in \{C, F\}$. C is for coarse scale and F stands for fine scale.

The vectors are then encoded by the sparse coding algorithm. They are encoded so that they can be represented by a linear combination of basis functions. The basis functions are picked from an over-complete dictionary $\phi^j(t) = \{\phi_n^j(t)\}_{n=1}^N$. In this research, there are 4 dictionaries. One pair for the stereo vision ($d = s$), and another for the motion parallax ($d = m$) as shown in Fig. 8.1. Each pair is responsible for coarse scale and fine scale. Initially randomized and normalized $N = 288$ basis functions are used to create the dictionaries. We use matching pursuit algorithm to find the sparse representation of the input vector with respect to the weighted sum

$$x_i^j(t) \approx \hat{x}_i^j(t) = \sum_{n=1}^{N} a_{i,n}^j(t)\phi_n^j(t) \ . \tag{8.1}$$

In this research, we limit the number of non-zero scalar coefficients $a_{i,n}(t)$ to be 10 elements to ensure the sparseness of the encoding (efficient coding). To be associated with

the reinforcement learner later, we define the pool activity which represent the activity of the neuron cells as follows:

$$f^j(t) = \begin{bmatrix} f_1^j(t) \\ f_2^j(t) \\ \vdots \\ f_P^j(t) \end{bmatrix} . \tag{8.2}$$

Where, each element of the vector $f^j(t)$ is described as:

$$f_n(t) = \sum_{i=1}^{P} a_{i,n}(t)^2. \tag{8.3}$$

The reconstruction error is defined as:

$$e(t) = \frac{1}{P} \sum_{i=1}^{P} \frac{\|x_i(t) - \sum_{n=1}^{N} a_{i,n}(t)\phi_n(t)\|^2}{\|x_i(t)^2\|}. \tag{8.4}$$

The error is used to update the dictionaries with the gradient descent method. Every update, the dictionaries are normalized.

8.2.2 Reinforcement Learning

Pooled activities from both coarse scale and fine scale are combined to represent the state $f_n(t)$ for the reinforcement learner as follows:

$$f(t) = \begin{bmatrix} f^C(t) \\ f^F(t) \end{bmatrix} . \tag{8.5}$$

The reward that is given to the learning agent is a negative of the summation of reconstruction error from both scales which is described as:

$$R_{d,k}(t) = -(e^C(t) + e^F(t)) . \tag{8.6}$$

Where, $k \in \{k_1, k_2\}$ and $d \in \{m, s\}$. m is for motion parallax. s is for stereo vision. An actor-critic algorithm number 3 proposed in [95] is employed for the leaner agent. For action selection, we use Gibbs distribution (softmax) for probabilistically choosing an action as follows:

$$\pi(f(t), a_t) = \frac{e^{z_a}}{\sum_{a' \in A} e^{z_{a'}}} . \tag{8.7}$$

For each action, the activation value z_a is given by:

$$z_a = \sum_{n=1}^{N} w_a(t) f_n(t) \ , \tag{8.8}$$

where $w_a(t)$ is a weight vector from the state $f(t)$ to action a that is initially random. The action is pan angle of the cameras in degrees. Possible actions a are contained in a set of actions A. In this research we use $A = \{-0.2°, -0.1°, -0.05°, 0°, 0.05°, 0.1°, 0.2°\}$. Thus, the policy maps $f(t)$ to $a \in A$. The selected actions are $P_{m,k_1}(t)$ for motion parallax and $P_{s,k_2}(t)$ for stereo vision.

8.2.3 Depth Representation

A feed forward artificial neural network is used to translate eye movements to the visual stimulus distance. The network consist of input, hidden, and output layer. The eye movements are stored for depth estimation in every iteration after the stereo vision were executed.

Amount of eye movements $\vec{q}$ are used to trained the neural network.

$$\vec{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{8.9}$$

where, q_1 represents the left eye's pan movement at leftmost position. q_2 is vergence eye movement at leftmost position. q_3 is left eye's pan movement at rightmost position. q_4 is vergence eye movement at rightmost position.

Levengerg-Marquardt method [102] is used for training the neural network. Hidden layer composes of 10 neurons which has sigmoid transfer function. Input layer is the vector $\vec{q}$. The output has one neurons which is the estimated depth supervised by ground truth depth provided by a supervisor.

8.3 Simulations & Results

8.3.1 Experimental Setup

We test multiple cues to estimate the distance of the object starting from the observer (robot). We use V-REP as the 3D environment simulator for the robot. The algorithm and framework are employed in MATLAB. The target distance can be varied in 0.1 meter interval starting from 1 to 3 meters. The absolute distance between the left and right position $\delta = 0.1$ meter. Baseline distance of the two cameras is 0.06 meter. We use $h = 30$ iterations. The object's texture is interchangeable with the prepared 100 textures.

In this simulation, we define the MAE as follows:

$$\mathrm{MAE(t)} = \frac{1}{1000} \sum_{\mathrm{k=0}}^{999} \left| \theta(\mathrm{t} + 29 + 30\mathrm{k}) - \theta^*(\mathrm{t} + 29 + 30\mathrm{k}) \right| . \tag{8.10}$$

Where, $\theta(t)$ is the pan/vergence angle of the eye and $\theta^*(t)$ is the expected pan/vergence angle.

8.3.2 Development of the Visual Dictionary

To analyze the distribution and variance of the visual dictionaries used in the sensory coding model, we apply principal component analysis (PCA). For the first stage of the visual dictionaries, it is expected to see each parts are redundant to each other. In Figs. 8.2a - 8.2c, they show the first and the second principal component of the visual dictionaries. In the final stage, we can see that the dictionaries are more sparsely distributed than the first stage.

8.3.3 Eye Movement Performance

We use the earlier defined MAE to measure the eye movement performance. The error compares the actual eye movements versus the expected eye movements. Fig. 8.3a represents the eye movement MAE.

While, Fig. 8.3b shows the depth perception performance. The eye movement information $\vec{q}$ which is effected by different experimental conditions is used as inputs for the

neural network. The depth estimation output is used to calculated the depth estimation MAE.

We can see that the robot could learn and improve both sensory coding and eye movement. In addition, the robot learns to estimate depth with stereo vision and motion parallax together.

8.3.4 Robustness Test

Since in practical application, disturbances are expected to happen. In this simulation, we test one of the robustness of the system. We apply a constant rotation to the dominant eye camera in roll-plane Fig. 8.4. As we can see in Fig. 8.4a, the eye movement MAE noticeably increased after the perturbation is applied (after the dashed line). Motion parallax cue was not effected much since the rotation effect both of the input images. However, for the stereo vision which depends on the motion parallax cue receive the effect much more. However, the framework could recover from the disturbance and reduce the MAE significantly for the stereo vision cue. Since depth perception is supported by both cues the depth estimation performance could be recovered similar to the performance before the test Fig. 8.4b.

8.4 Summary

In this chapter, we proposed a novel developmental learning framework to actively the active depth perception during self-induced lateral body movements. The proposed framework can simultaneously develop the sensory representation, eye movement control and integration of the visual depth cues such as stereo disparity and motion parallax. In order to avoid the conflict of multiple eye movements, the two different eye movements are sequentially trained and generated, while they share the same learning architecture. Finally, the generated multiple eye movements are effectively used to represent the depth information. Also, the proposed learning framework can be seamlessly recovered from the external perturbations.

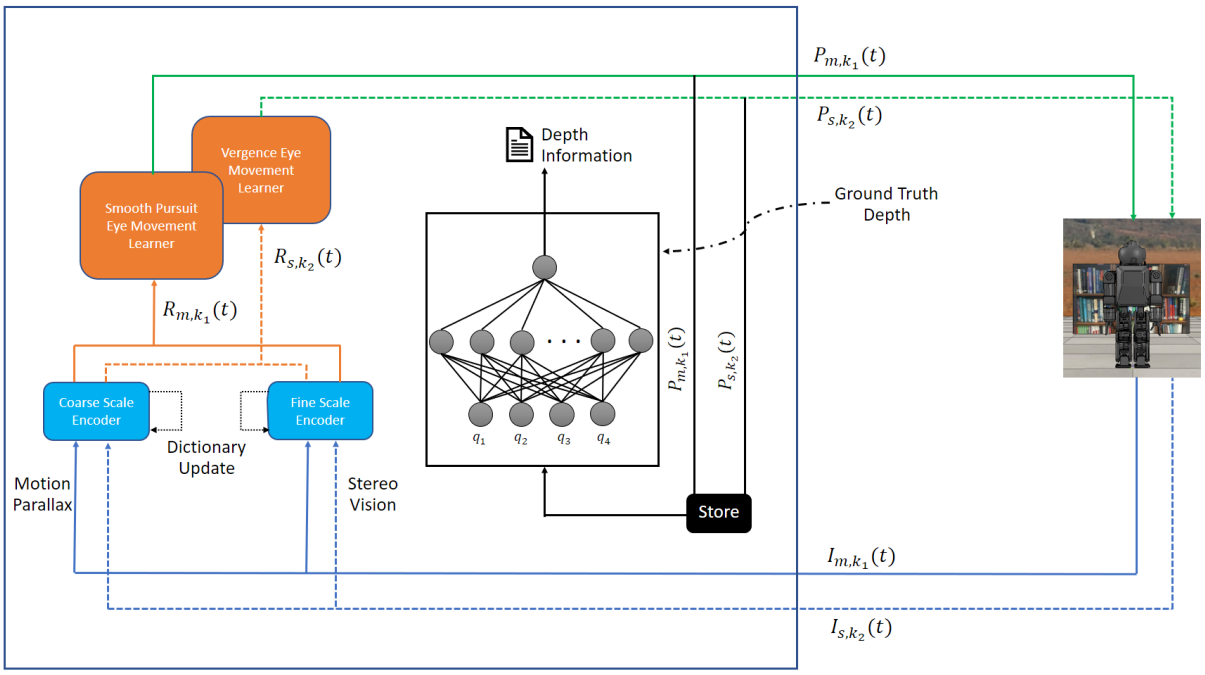
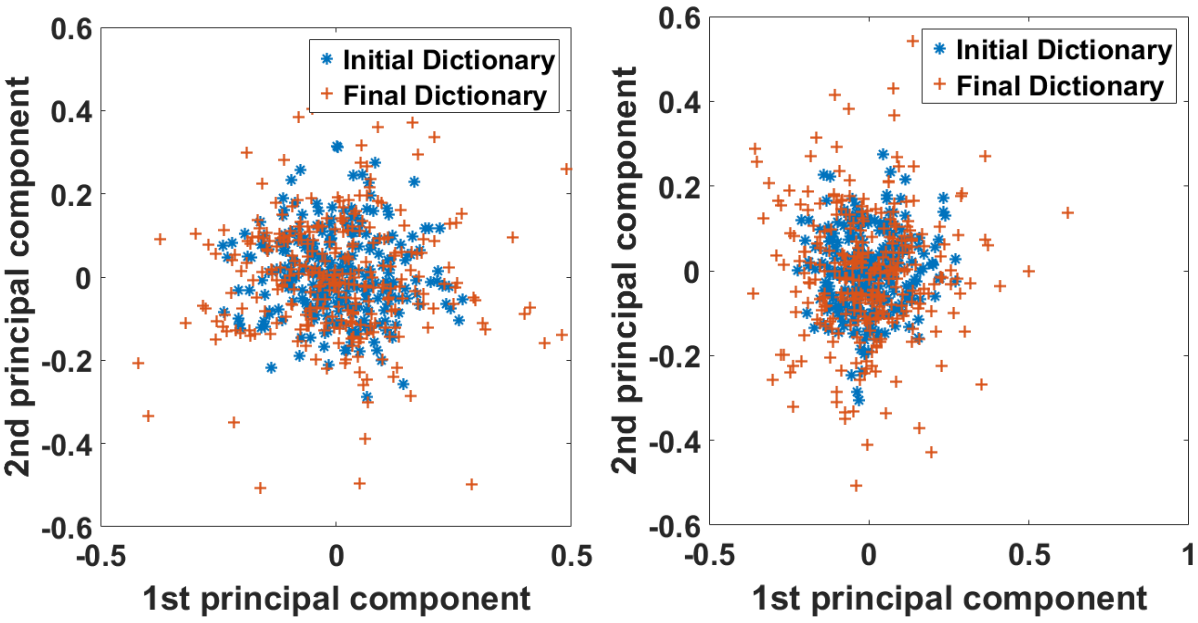
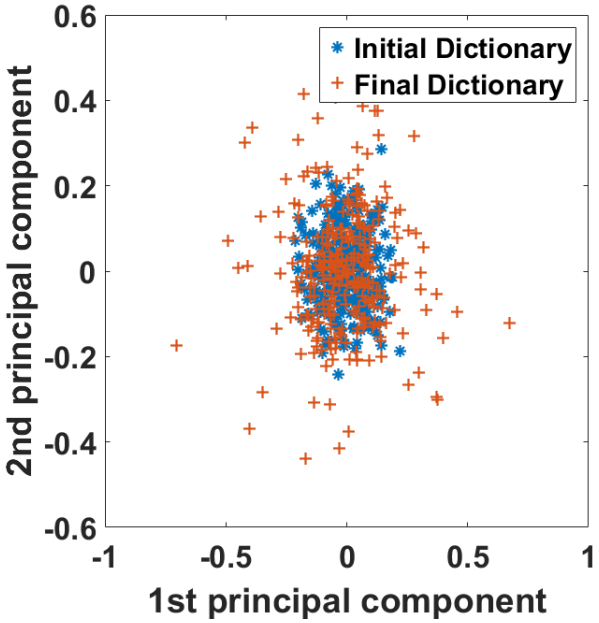


Figure 8.1: Model architecture. (1) At the first step k_1, to perform the motion parallax, the robot captures the successive images $I_{m,k1}(t)$ during the self-induced lateral body movement which are fed into the sensory encoders with multiple image scales. Later, an output reward signal, $R_{m,k_1}(t)$, is sent to the reinforcement learner to generate an appropriate eye movement to hold the fixation during the body movement. Finally, pan command $P_{m,k_1}(t)$ is sent to the robot and it generates the smooth pursuit eye movement for dominant eye camera to maximize the redundancy between the successive images. (2) At the second step k_2, stereo images $I_{s,k_2}(t)$ are captured from both two cameras and sent to the sensory encoders. An output reward signal, $R_{s,k_2}(t)$, is sent to the reinforcement learner to generate the vergence command $P_{s,k_2}(t)$ to maximize the redundancy between the stereo images. The visual dictionaries are then updated based on visual reconstruction errors for both of visual depth cues. Finally, the stored eye movements (q_1, q_2, q_3, and q_4) are used as an input for the neural network to represent the depth information which is given by human-robot interaction.

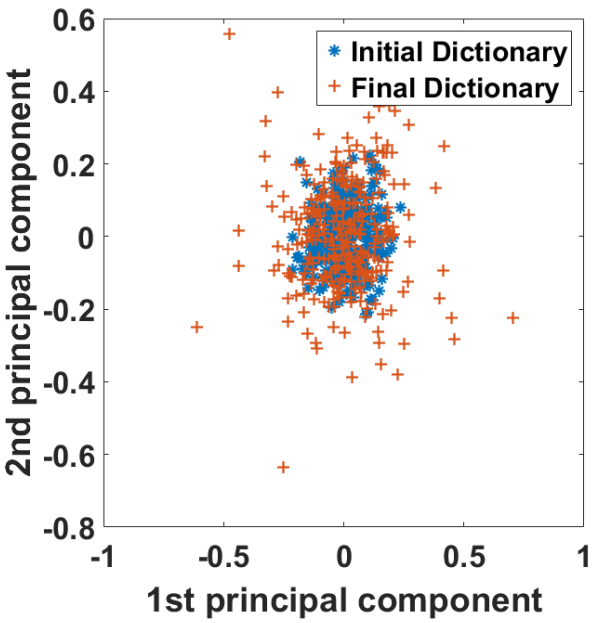


(a) Coarse scale for stereo disparity

(b) Fine scale for stereo disparity

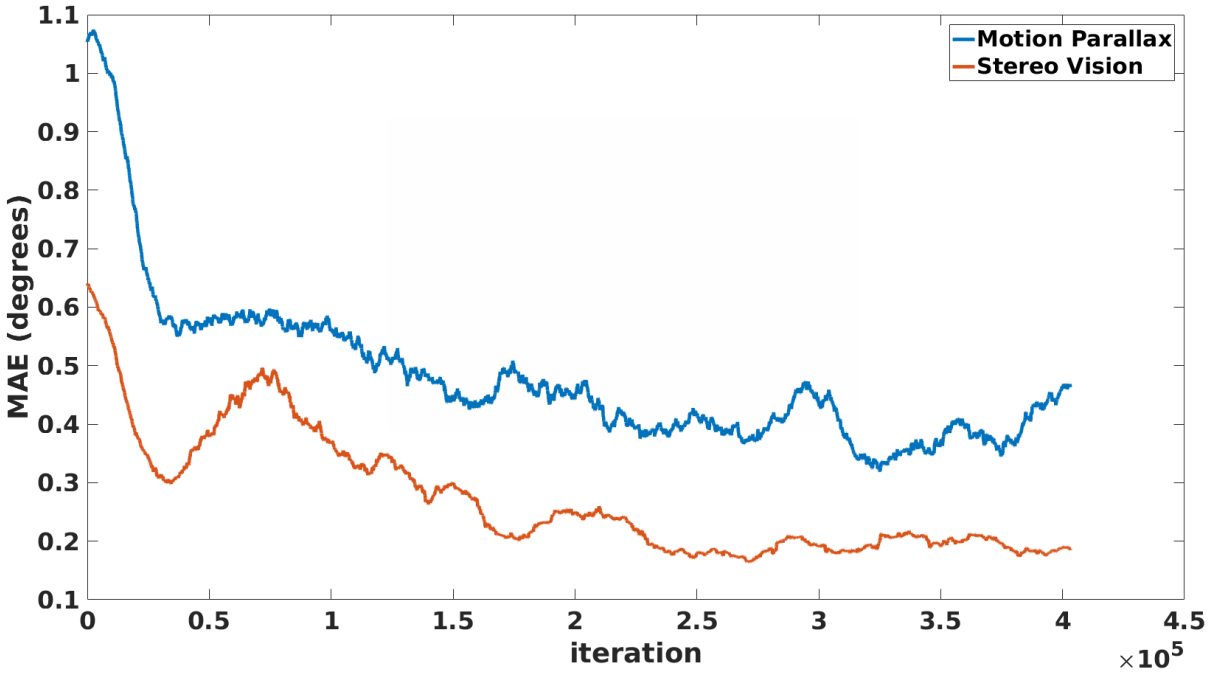


(c) Coarse scale for motion parallax

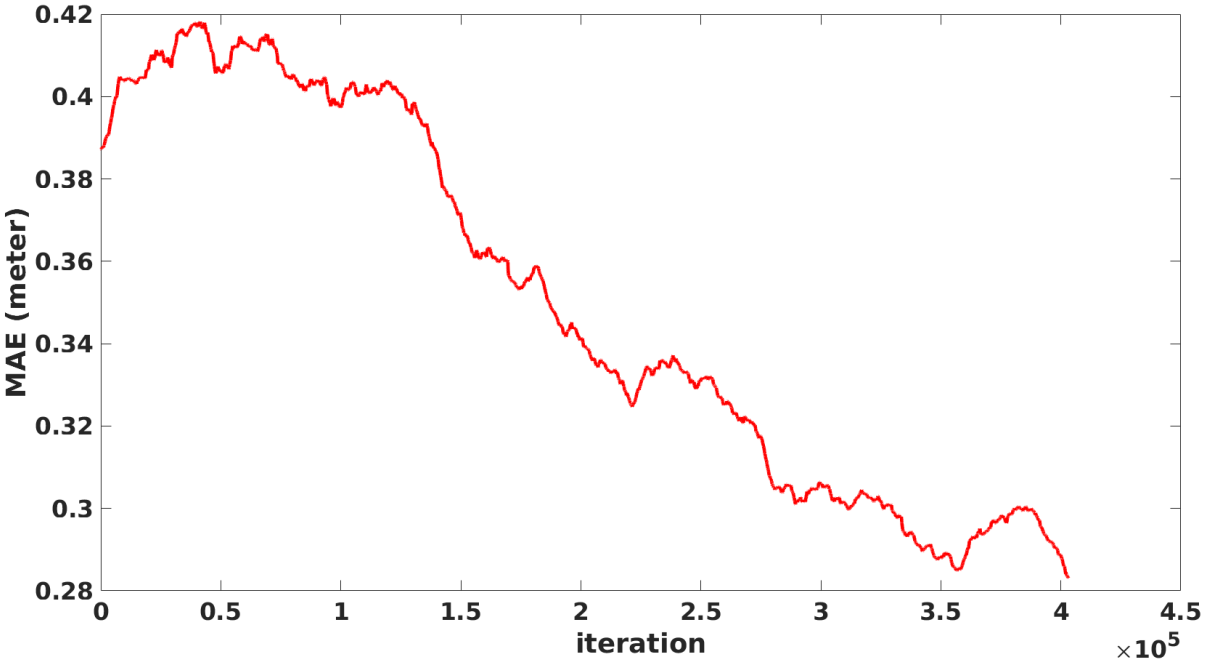


(d) Fine scale for motion parallax

Figure 8.2: Visualization of development of the visual dictionaries. The distribution of the visual dictionaries using the first and second PCs at the initial time and the end of training, respectively.

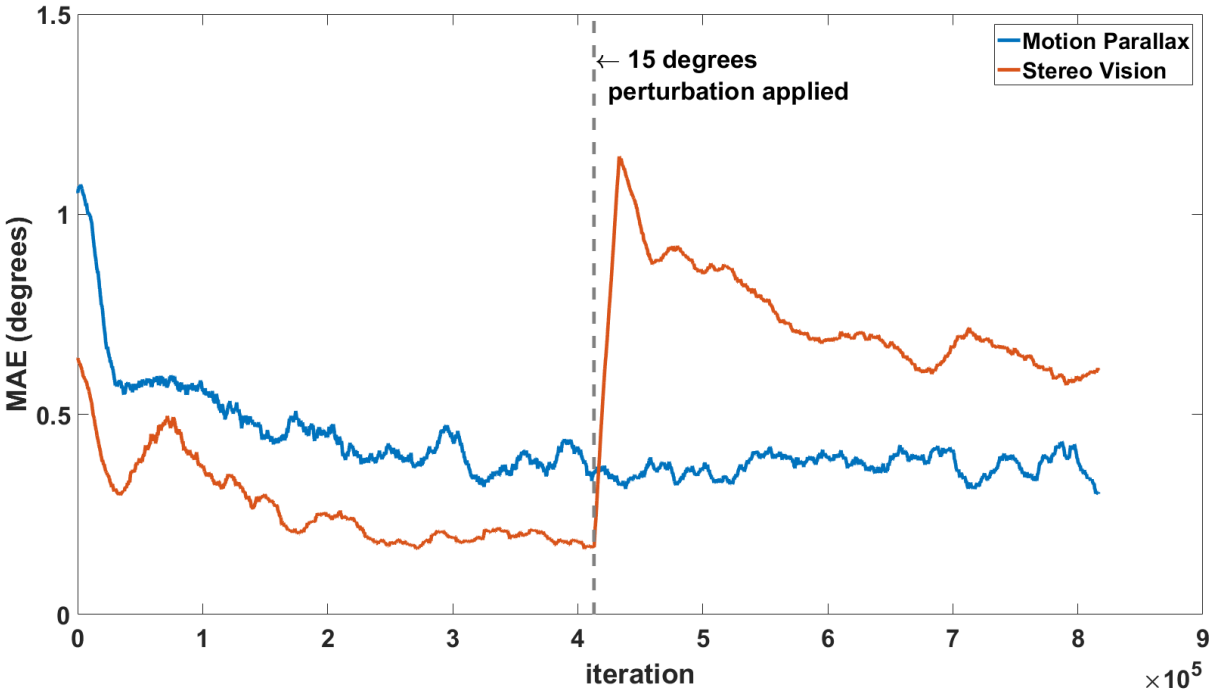


(a) MAE of eye movements

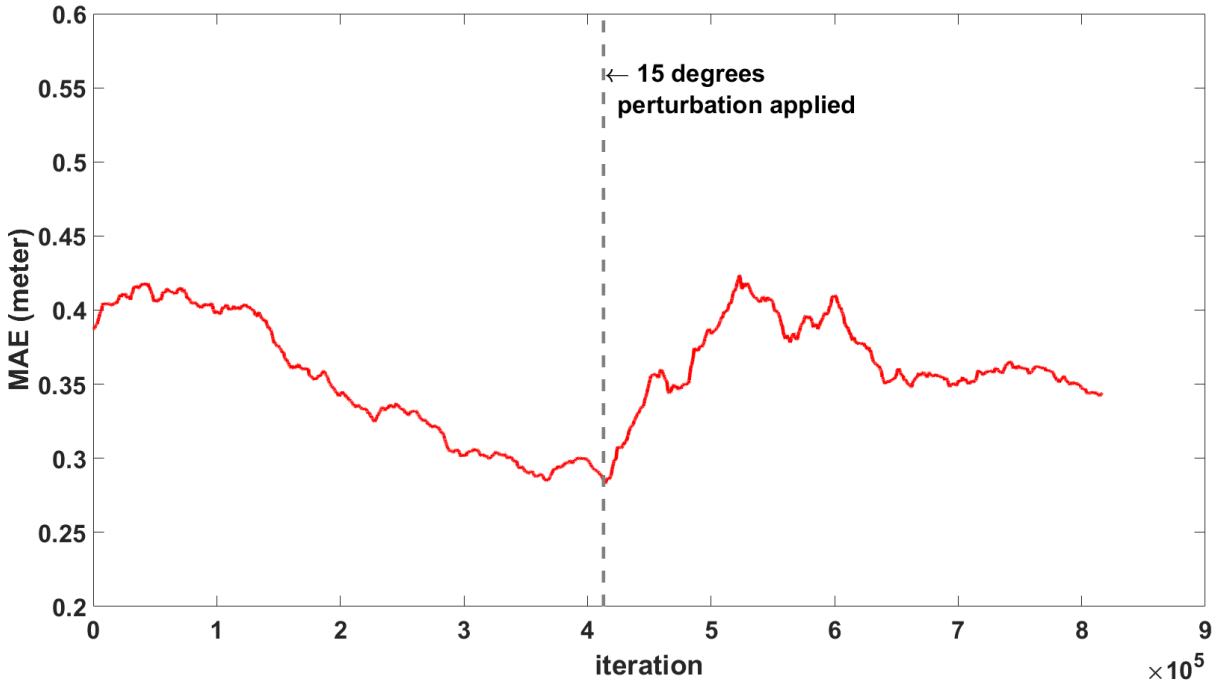


(b) MAE of depth perception

Figure 8.3: The development of the system. visual representation (coding), eye movement and depth estimation. a represents the eye movement MAE. b shows depth estimation MAE.



(a) MAE of eye movements with perturbation



(b) MAE of depth perception with perturbation

Figure 8.4: Robustness test. a eye movement MAE after perturbation. b MAE of the depth perception after perturbation

Chapter 9

Conclusions

This last chapter summarizes all of the works done and emphasizes its contributions to the cognitive developmental robotics as well as the other research fields. Since the ultimate goal of implementing a full active depth perception has yet to be achieved, this chapter also discusses the room for the future improvement.

9.1 Summary

To create the biological inspired active depth perception model, it needs to be developmental and has the action-perception cycle. Most of the studies either consider the action and perception separately or do not have the developmental learning ability.

In this research we proposed a frameworks that have those properties. We use the active efficient coding together with the reinforcement learning to create a tight connection between action and perception. The depth perception module has been developed with the artificial neural networks. These parts are the keys to implement the goal model.

Throughout the research the proposed frameworks are verified and analyzed with simulations and experiments. The finding and the framework related to each framework can be found in the summary section in each chapter. The contributions of this research will be discussed in the next section.

Importantly, this research does not focus to compete with the other computer vision techniques that is designed and optimized to solve a specific problem (such as depth estimation performance). But, this research aims to prove the concept of creating the

biological inspired model that can learn by itself. Improvement of the performance maybe done in the future works.

9.2 Contributions

The contributions of this research can be seen from different viewpoints. The two fields that mainly concern the research would be robotics and neural science. For robotics, the developed framework could be implemented in the biological inspired robots such as humanoid robot which is continuously developing by many researchers. It will enable the developmental learning scheme for the robot. As for the neural science, the framework could also be used as the model for the development in various vision pathologies or under unnatural rearing conditions. This could enable a new forms of clinical intervention.

To summarize the unique and novel points, they are listed as follows.

1. We proposed a novel framework that can generate smooth pursuit eye movement to fixate an object during the self-induced lateral movement. It is also able to estimate the distance between the robot and the fixating object with supervised learning. The framework is developmental learning which means it can learn and adapt to the changes and new things in its configuration and the environment.

2. We also proposed a new learning strategy to enable the framework improve the depth estimation accuracy and step beyond its limitation.

3. The framework can select an optimal lateral movement that is require to distinguish between two close depths. This enable the natural movement instead of the predefined movement.

4. Finally, the two cues of the active depth perception which are stereo vision and motion parallax are integrated together into a single unified framework.

To the best of our knowledge, no study has attempted to propose the active depth perception frameworks for developmental robots under the efficient coding theory. This approach enables robots not only to autonomously learn sensory representation and eye movement controls but also the first step toward creating active depth perception.

9.3 Future Work

In this research, we have the success on implementing and creating the frameworks. However, our final goal has not been reached yet. Here we discuss some of the possible improvement.

1. **Optic Flow Extension**

   As we have mentioned before that there are three types of depth perception which are stereo vision, motion parallax, and optic flow. We have studied the stereo vision framework. We have proposed a way to extended the active depth perception with motion parallax. However, there is one remaining type of depth perception that have not yet been used yet which is optic flow. To perceive depth by optic flow, we have to move forward and backward. When we are moving forward and backward, we can sense that the closer object have the size increased more than the object that is far away. So, the question is left that can we utilize those information to extend the framework.

   The change of the size of the perceived visual stimuli due to the proximity between the observer and the object could be considered in the sensory coding model. Smooth pursuit eye movement (tilt) could be introduced in the reinforcement learning.

2. **Active Depth Perception Integration**

   The prospective of this thesis is to mimic the depth perception system in developed organisms. We have shown that it is possible to integrate both the stereo vision and motion parallax together. With the implementation of the optic flow, the complete integration of the active perception could be done. We may use the same dominant eye concept to additionally include the optic flow, since the optic flow is also the smooth pursuit eye movement but in the different rotational axis.

3. **Multiple Visual Stimuli**

   Even though, we can extended the active depth perception part with motion parallax and estimate the depth, we can only find the distance of a single object in the scene. A visual attention or saliency map could be added to the framework to

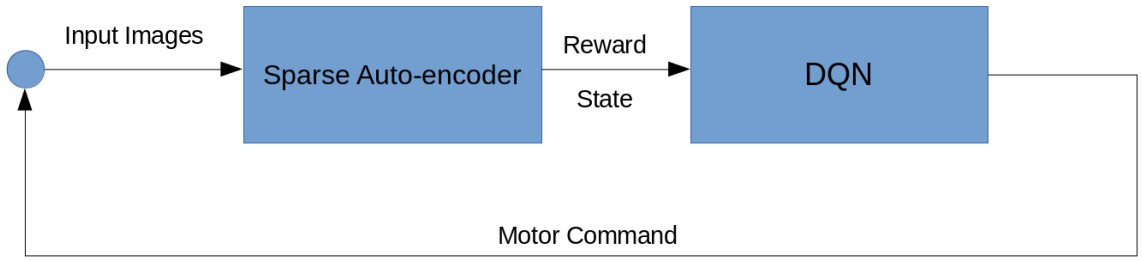


Figure 9.1: A candidate model that are designed by using deep-learning studies

enable fixating in the various area in the field of view. This should also enable the ability of observing multiple egocentric and allocentric distance of objects in an environment.

4. **Unsupervised Depth Estimation**

   In this research, we considered only the supervised learning method which is the artificial neural network. The frameworks need supervised signals in order to learn the depth in the form of metric system. However, an unsupervised learning method may be used to let the framework learn the depth in the form of its own unit. For an example, after fixating an object, the robot may move to touch the object and then associate the number of step it took with the amount of eye movement.

5. **Deep Learning Infusion**

   With the recent developments, deep learning has become popular in the machine learning field. With the state-of-the-art deep-learning perspective, we can employ a sparse auto-encoder as a visual leaner, while a deep Q-network (DQN) [111] is used as an action learner. The model maybe described as shown in Figure 9.1 below.

   According to the previous works, if a sensory coding is defined and it could find a meaningful representation of the input images, then a reinforcement learning algorithm that is capable of mapping action-value function in the continuous state could be used to achieve the same task. Even though the performance of the candidate model is unknown, but we can expected a better performance with the cutting-edge sparse auto-encoder and DQN framework.

Bibliography

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.

[3] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller, and Y. LeCun, "Deep belief net learning in a long-range vision system for autonomous off-road driving," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on.* IEEE, 2008, pp. 628–633.

[4] C. Becker-Asano and H. Ishiguro, "Evaluating facial displays of emotion for the android robot geminoid f," in *Affective Computational Intelligence (WACI), 2011 IEEE Workshop on.* IEEE, 2011, pp. 1–8.

[5] F. Tanaka, K. Isshiki, F. Takahashi, M. Uekusa, R. Sei, and K. Hayashi, "Pepper learns together with children: Development of an educational application," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on.* IEEE, 2015, pp. 270–275.

[6] B.-J. Lee, J. Choi, K. Park, C. Lee, S. Choi, C. Han, D.-S. Han, C. Baek, P. Emaase, and B. Zhang, "Perception-action-learning system for mobile social-service robots using deep learning," *AAAI-2018 Demos Track, New Orleans*, 2018.

[7] J. Gerbscheid, T. Groot, and A. Visser, "Intelligent news conversation with the pepper robot," in *Proceedings of the 29th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC 2017), Groningen, The Netherlands*, 2017.

[8] A. Chibani, Y. Amirat, S. Mohammed, E. Matson, N. Hagita, and M. Barreto, "Ubiquitous robotics: Recent challenges and future trends," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1162–1172, 2013.

[9] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, F. Nori, L. Fadiga, B. Wrede, K. Rohlfing, E. Tuci, K. Dautenhahn, J. Saunders, and A. Zeschel, "Integration of action and language knowledge: A roadmap for developmental robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 167–195, Sept 2010.

[10] P. Gomes, "Surgical robotics: Reviewing the past, analysing the present, imagining the future," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 2, pp. 261–266, 2011.

[11] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robotics and Autonomous Systems*, vol. 37, no. 2, pp. 185–193, 2001.

[12] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.

[13] P. Maes, *Designing autonomous agents: Theory and practice from biology to engineering and back*. MIT press, 1990.

[14] R. A. Brooks, "Intelligence without representation," *Artificial intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.

[15] P. E. Agre, "Computational research on interaction and agency," *Artificial intelligence*, vol. 72, no. 1-2, pp. 1–52, 1995.

[16] R. Pfeifer and C. Scheier, *Understanding intelligence*. MIT press, 2001.

[17] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[18] G. Sandini, G. Metta, and D. Vernon, "Robotcub: An open framework for research in embodied cognition," in *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, vol. 1. IEEE, 2004, pp. 13–32.

[19] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *IEEE transactions on evolutionary computation*, vol. 11, no. 2, pp. 151–180, 2007.

[20] S. Fuke, M. Ogino, and M. Asada, "Body image constructed from motor and tactile images with visual information," *International Journal of Humanoid Robotics*, vol. 4, no. 02, pp. 347–364, 2007.

[21] C. Nabeshima, M. Lungarella, and Y. Kuniyoshi, "Timing-based model of body schema adaptation and its role in perception and tool use: A robot case study," in *Development and Learning, 2005. Proceedings. The 4th International Conference on.* IEEE, 2005, pp. 7–12.

[22] Y. Yoshikawa, H. Kawanishi, M. Asada, and K. Hosoda, "Body scheme acquisition by cross modal map learning among tactile, visual, and proprioceptive spaces," 2002.

[23] Y. Yoshikawa, "Subjective robot imitation by finding invariance," 2005.

[24] A. Stoytchev, L. Berthouze, C. Prince, M. Littman, H. Kozima, and C. Balkenius, "Toward video-guided robot behaviors," in *Proceedings of the Seventh International Conference on Epigenetic Robotics (EpiRob)*, vol. 135, 2007, pp. 165–172.

[25] M. Hersch, E. Sauser, and A. Billard, "Online learning of the body schema," *International Journal of Humanoid Robotics*, vol. 5, no. 02, pp. 161–181, 2008.

[26] J.-R. Duhamel, C. L. Colby, and M. E. Goldberg, "Ventral intraparietal area of the macaque: congruent visual and somatic response properties," *Journal of neurophysiology*, vol. 79, no. 1, pp. 126–136, 1998.

[27] M. S. Graziano and D. F. Cooke, "Parieto-frontal interactions, personal space, and defensive behavior," *Neuropsychologia*, vol. 44, no. 6, pp. 845–859, 2006.

[28] M. I. Sereno and R.-S. Huang, "A human parietal face area contains aligned head-centered visual and tactile maps," *Nature neuroscience*, vol. 9, no. 10, p. 1337, 2006.

[29] A. Klin, W. Jones, R. Schultz, F. Volkmar, and D. Cohen, "Defining and quantifying the social phenotype in autism," *American Journal of Psychiatry*, vol. 159, no. 6, pp. 895–908, 2002.

[30] M. Ogino, A. Watanabe, and M. Asada, "Detection and categorization of facial image through the interaction with caregiver," in *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on.* IEEE, 2008, pp. 244–249.

[31] N. Otsu and T. Kurita, "A new scheme for practical flexible and intelligent vision systems." in *MVA*, 1988, pp. 431–435.

[32] H. Kozima, C. Nakagawa, and H. Yano, "Attention coupling as a prerequisite for social interaction," in *Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003. The 12th IEEE International Workshop on.* Ieee, 2003, pp. 109–114.

[33] Y. Nagai, M. Asada, and K. Hosoda, "Learning for joint attention helped by functional development," *Advanced Robotics*, vol. 20, no. 10, pp. 1165–1181, 2006.

[34] J. Triesch, C. Teuscher, G. O. Deák, and E. Carlson, "Gaze following: why (not) learn it?" *Developmental science*, vol. 9, no. 2, pp. 125–147, 2006.

[35] F. Kaplan and V. V. Hafner, "The challenges of joint attention," *Interaction Studies*, vol. 7, no. 2, pp. 135–169, 2006.

[36] D. A. Baldwin, "Infants' contribution to the achievement of joint reference," *Child development*, vol. 62, no. 5, pp. 874–890, 1991.

[37] T. Schreiber, "Measuring information transfer," *Physical review letters*, vol. 85, no. 2, p. 461, 2000.

[38] Y. Yoshikawa, T. Nakano, M. Asada, and H. Ishiguro, "Multimodal joint attention through cross facilitative learning based on μx principle," in *Development and*

Learning, 2008. ICDL 2008. 7th IEEE International Conference on. IEEE, 2008, pp. 226–231.

[39] G. Westermann and E. R. Miranda, "A new model of sensorimotor coupling in the development of speech," *Brain and language*, vol. 89, no. 2, pp. 393–400, 2004.

[40] H. Kanda, T. Ogata, K. Komatani, and H. G. Okuno, "Segmenting acoustic signal with articulatory movement using recurrent neural network for phoneme acquisition," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on.* IEEE, 2008, pp. 1712–1717.

[41] J. Hornstein and J. Santos-Victor, "A unified approach to speech production and recognition based on articulatory motor representations," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on.* IEEE, 2007, pp. 3442–3447.

[42] T. Hashimoto, M. Senda, and H. Kobayashi, "Realization of realistic and rich facial expressions by face robot," in *Robotics and Automation, 2004. TExCRA'04. First IEEE Technical Exhibition Based Conference on.* IEEE, 2004, pp. 37–38.

[43] H. Papousek, "Intuitive parenting: A dialectic counterpart to the infant's integrative competence," *Handbook of infant development*, 1987.

[44] K. Ishiguro, N. Otsu, and Y. Kuniyoshi, "Inter-modal learning and object concept acquisition." in *MVA*, 2005, pp. 148–151.

[45] L. Steels and F. Kaplan, "Aibos first words: The social learning of language and meaning," *Evolution of communication*, vol. 4, no. 1, pp. 3–32, 2000.

[46] N. Iwahashi, "Language acquisition through a human–robot interface by combining speech, visual, and behavioral information," *Information Sciences*, vol. 156, no. 1-2, pp. 109–121, 2003.

[47] Y. Zhao, C. Rothkopf, J. Triesch, and B. Shi, "A unified model of the joint development of disparity selectivity and vergence control," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, Nov 2012, pp. 1–6.

[48] L. Lonini, Y. Zhao, P. Chandrashekhariah, B. Shi, and J. Triesch, "Autonomous learning of active multi-scale binocular vision," in *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*, Aug 2013, pp. 1–6.

[49] T. Vikram, C. Teuliere, C. Zhang, B. Shi, and J. Triesch, "Autonomous learning of smooth pursuit and vergence through active efficient coding," in *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*. IEEE, 2014, pp. 448–453.

[50] C. Teulière, S. Forestier, L. Lonini, C. Zhang, Y. Zhao, B. Shi, and J. Triesch, "Self-calibrating smooth pursuit through active efficient coding," *Robotics and Autonomous Systems*, vol. 71, pp. 3–12, 2015.

[51] C. Zhang, Y. Zhao, J. Triesch, and B. E. Shi, "Intrinsically motivated learning of visual motion perception and smooth pursuit," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1902–1908.

[52] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[53] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and autonomous systems*, vol. 42, no. 3-4, pp. 143–166, 2003.

[54] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.

[55] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr, "Urban 3d semantic modelling using stereo vision," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 580–585.

[56] X. Kuang, M. Gibson, B. E. Shi, and M. Rucci, "Active vision during coordinated head/eye movements in a humanoid robot," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1423–1430, 2012.

[57] M. Antonelli, A. P. Del Pobil, and M. Rucci, "Depth estimation during fixational head movements in a humanoid robot," in *International Conference on Computer Vision Systems*. Springer, 2013, pp. 264–273.

[58] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in neural information processing systems*, 2006, pp. 1161–1168.

[59] H. Zhuang, R. Sudhakar, and J.-y. Shieh, "Depth estimation from a sequence of monocular images with known camera motion," *Robotics and autonomous systems*, vol. 13, no. 2, pp. 87–95, 1994.

[60] S. Jeong, S.-W. Ban, and M. Lee, "Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment," *Neural networks*, vol. 21, no. 10, pp. 1420–1430, 2008.

[61] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *European Conference on Computer Vision*. Springer, 2002, pp. 661–675.

[62] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 213–228, 2015.

[63] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International journal of computer vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[64] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1822–1829.

[65] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1838–1845.

[66] H. Lee, S. Jeong, and N. Y. Chong, "Unsupervised learning approach to attention-path planning for large-scale environment classification," in *Intelligent Robots and*

Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014, pp. 1447–1452.

[67] M. Antonelli, A. P. Del Pobil, and M. Rucci, "Bayesian multimodal integration in a robot replicating human head and eye movements," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on.* IEEE, 2014, pp. 2868–2873.

[68] X. Kuang, M. Gibson, B. E. Shi, and M. Rucci, "Active vision during coordinated head/eye movements in a humanoid robot," *Robotics, IEEE Transactions on*, vol. 28, no. 6, pp. 1423–1430, 2012.

[69] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning.* ACM, 2005, pp. 593–600.

[70] A. Saxena, J. Schulte, A. Y. Ng, *et al.*, "Depth estimation using monocular and stereo cues." in *IJCAI*, vol. 7, 2007.

[71] K. F. MacDorman, "Grounding symbols through sensorimotor integration," *Journal of the Robotics Society of Japan*, vol. 17, no. 1, pp. 20–24, 1999.

[72] J. Law, P. Shaw, and M. Lee, "A biologically constrained architecture for developmental learning of eye–head gaze control on a humanoid robot," *Autonomous Robots*, vol. 35, no. 1, pp. 77–92, 2013.

[73] E. G. Freedman, "Coordination of the eyes and head during visual orienting," *Experimental brain research*, vol. 190, no. 4, p. 369, 2008.

[74] N. Chumerin, A. Gibaldi, S. P. Sabatini, and M. M. Van Hulle, "Learning eye vergence control from a distributed disparity representation," *International journal of neural systems*, vol. 20, no. 04, pp. 267–278, 2010.

[75] B. J. Grzyb, V. Castelló, M. Antonelli, and A. P. Del Pobil, "Integration of static and self-motion-based depth cues for efficient reaching and locomotor actions," in *International Conference on Artificial Neural Networks.* Springer, 2012, pp. 322–329.

[76] M. Asada, E. Uchibe, and K. Hosoda, "Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development," *Artificial Intelligence*, vol. 110, no. 2, pp. 275–292, 1999.

[77] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[78] C.-Y. Tsai, C.-C. Wong, C.-J. Yu, C.-C. Liu, and T.-Y. Liu, "A hybrid switched reactive-based visual servo control of 5-dof robot manipulators for pick-and-place tasks," *IEEE Systems Journal*, vol. 9, no. 1, pp. 119–130, 2015.

[79] Y. Wang, H. Lang, and C. W. de Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots," *IEEE/ASME transactions on Mechatronics*, vol. 15, no. 5, pp. 757–769, 2010.

[80] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev*, pp. 183–193, 1954.

[81] H. B. Barlow, *Possible principles underlying the transformation of sensory messages*. Cambridge, MA: MIT Press, 1961.

[82] D. J. Field, "What is the goal of sensory coding?" *Neural Comput.*, vol. 6, no. 4, pp. 559–601, July 1994.

[83] A. Gibaldi, A. Canessa, F. Solari, and S. P. Sabatini, "Autonomous learning of disparity–vergence behavior through distributed coding and population reward: Basic mechanisms and real-world conditioning on a robot stereo head," *Robotics and Autonomous Systems*, vol. 71, pp. 23–34, 2015.

[84] B. Kuipers, P. Beeson, J. Modayil, and J. Provost, "Bootstrap learning of foundational representations," *Connection Science*, vol. 18, no. 2, pp. 145–158, June 2006.

[85] J. Mugan and B. Kuipers, "Autonomous learning of high-level states and actions in continuous environments," *Autonomous Mental Development, IEEE Transactions on*, vol. 4, no. 1, pp. 70–86, March 2012.

[86] J. Weng and M. Luciw, "Brain-like emergent spatial processing," *Autonomous Mental Development, IEEE Transactions on*, vol. 4, no. 2, pp. 161–185, June 2012.

[87] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Research*, vol. 37, no. 23, pp. 3311 – 3325, 1997.

[88] M. Wexler and J. J. Van Boxtel, "Depth perception by the active observer," *Trends in cognitive sciences*, vol. 9, no. 9, pp. 431–438, 2005.

[89] F. Toates, "Vergence eye movements," *Documenta Ophthalmologica*, vol. 37, no. 1, pp. 153–214, 1974.

[90] D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proc. R. Soc. Lond. B*, vol. 204, no. 1156, pp. 301–328, 1979.

[91] B. Rogers and M. Graham, "Motion parallax as an independent cue for depth perception," *Perception*, vol. 8, no. 2, pp. 125–134, 1979.

[92] E. J. Gibson, J. J. Gibson, O. W. Smith, and H. Flock, "Motion parallax as a determinant of perceived depth." *Journal of experimental psychology*, vol. 58, no. 1, p. 40, 1959.

[93] D. N. Lee, "The optic flow field: The foundation of vision," *Phil. Trans. R. Soc. Lond. B*, vol. 290, no. 1038, pp. 169–179, 1980.

[94] J. J. Koenderink, "Optic flow," *Vision research*, vol. 26, no. 1, pp. 161–179, 1986.

[95] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor–critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[96] J. Semmlow, G. Hung, and K. Ciuffreda, "Quantitative assessment of disparity vergence components," *Investigative ophthalmology and visual science*, vol. 27, no. 4, p. 558564, April 1986.

[97] K. Stroyan and M. Nawrot, "Visual depth from motion parallax and eye pursuit," *Journal of mathematical biology*, vol. 64, no. 7, pp. 1157–1188, 2012.

[98] S. H. Ferris, "Motion parallax and absolute distance." *Journal of experimental psychology*, vol. 95, no. 2, p. 258, 1972.

[99] M. E. Ono, J. Rivest, and H. Ono, "Depth perception as a function of motion parallax and absolute-distance information." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 12, no. 3, p. 331, 1986.

[100] W. E. Vinje and J. L. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision," *Science*, vol. 287, no. 5456, pp. 1273–1276, 2000.

[101] H. Zhou, H. S. Friedman, and R. Von Der Heydt, "Coding of border ownership in monkey visual cortex," *Journal of Neuroscience*, vol. 20, no. 17, pp. 6594–6611, 2000.

[102] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*. Springer, 1978, pp. 105–116.

[103] M. Nawrot and K. Stroyan, "The motion/pursuit law for visual depth perception from motion parallax," *Vision research*, vol. 49, no. 15, pp. 1969–1978, 2009.

[104] K. Stroyan and M. Nawrot, "Visual depth from motion parallax and eye pursuit," *Journal of mathematical biology*, vol. 64, no. 7, pp. 1157–1188, 2012.

[105] J. Holmin and M. Nawrot, "Motion parallax thresholds for unambiguous depth perception," *Vision research*, vol. 115, pp. 40–47, 2015.

[106] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[107] E. Shneor and S. Hochstein, "Eye dominance effects in feature search," *Vision research*, vol. 46, no. 25, pp. 4258–4269, 2006.

[108] ——, "Eye dominance effects in conjunction search," *Vision research*, vol. 48, no. 15, pp. 1592–1602, 2008.

[109] J. Johansson, G. Ö. Seimyr, and T. Pansell, "Eye dominance in binocular viewing conditions," *Journal of vision*, vol. 15, no. 9, pp. 21–21, 2015.

[110] J. Frey and D. L. Ringach, "Binocular eye movements evoked by self-induced motion parallax," *The Journal of Neuroscience*, vol. 31, no. 47, pp. 17 069–17 073, 2011.

[111] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

Publications

Journal

[1] Tanapol Prucksakorn, Sungmoon Jeong, and Nak Young Chong, "A Self-Trainable Depth Perception Method from Eye Pursuit and Motion Parallax," Robotics and Autonomous Systems (2018) Vol. 109, pp. 27-37.

International Conference

[2] Tanapol Prucksakorn, Sungmoon Jeong, and Nak Young Chong, "Joint learning for smooth pursuit eye movement and moton parallax through active efficient coding," in Ubiquitous Robots and Ambient Intelligence (URAI), 2015 12th International Conference on (pp. 458-459). IEEE.

[3] Tanapol Prucksakorn, Sungmoon Jeong, Jochen Triesch, Hosun Lee, and Nak Young Chong, "Self-calibrating active depth perception via motion parallax," in Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2016 Joint IEEE International Conference on (pp. 103-108). IEEE

[4] Tanapol Prucksakorn, Sungmoon Jeong, and Nak Young Chong, "A Joint Learning Framework of Visual Sensory Representation, Eye Movements and Depth Representation for Developmental Robotic Agents," in International Conference on Neural Information Processing (pp. 867-876). Springer, Cham.