

AUTONOMOUS GROUNDING OF THE OPTICAL FLOW DETECTORS IN A
SIMULATED VISUOMOTOR SYSTEM OF THE FLY USING BEHAVIORALLY
MEANINGFUL ACTIONS

A Thesis

by

AMEY PARULKAR

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Chair of Committee,	Yoonsuck Choe
Committee Members,	Tracy A. Hammond
	John Buchanan
Head of Department,	Dilma Da Silva

August 2015

Major Subject: Computer Science

Copyright 2015 Amey Parulkar

ABSTRACT

The fly visual system, although tiny when compared to the mammalian visual system, can still perform highly sophisticated functions to help the animal navigate through the environment. An example of such a function is the detection of optical flow. Perceiving and responding to the optical flow is critical to the animal's survival and success. Higher level neurons in the fly's visual system, the Lobula Plate Tangential Cells (LPTCs), respond to different types of optical flow due to rotation and translation, by pooling information from elementary motion detectors (EMDs) in the lower level. In this sense, neuronal responses (spikes) from these optical flow detectors in the fly carry highly encoded signals.

In this thesis, I investigate how such highly encoded signals can be interpreted and utilized in the fly's brain, while solely operating on the internal spike patterns in its own brain. This is also called the grounding problem. With a computational model of the optical flow detectors in the fly, I show that action (or coordinated motor output) is the only way that the fly can learn the meaning of its internal spikes and generate meaningful, relevant behavior. I show this with two methods of integrating local optical flow information based on physiological and functional model of fly's LPTCs.

The results show that the proposed agent model based on fly's visuomotor system is able to learn a good sensory state to action mapping. This indicates that the agent model and the learning algorithm are able to give promising results for the motion grounding problem in synthetic and natural scenarios.

DEDICATION

Dedicated to my parents - Girish Parulkar and Vasanti Parulkar.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Prof. Yoonsuck Choe who has been a constant source of support and knowledge, for all his efforts in helping me write a research paper and this thesis and above all for being a great mentor and inspiring me to do good research. I would also like to thank Dr. Tracy Hammond and Dr. John Buchanan for their constant support throughout the research process.

NOMENCLATURE

EMD	Elementary Motion Detector
LPTC	Lobula Plate Tangential Cell
HS	Horizontal System
VS	Vertical System
mHS	Modeled Horizontal System
mVS	Modeled Vertical System
RYRL	Rotation Yaw Right to Left
RYLR	Rotation Yaw Left to Right
RPUD	Rotation Pitch Up to Down
RPDU	Rotation Pitch Down to Up
RRCL	Rotation Roll Clockwise
RRAC	Rotation Roll Anti-Clockwise
TLRI	Translation Radiate In
TLRO	Translation Radiate Out

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
NOMENCLATURE	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
LIST OF TABLES	xiii
1. INTRODUCTION	1
1.1 Overview	1
1.2 Background	2
1.3 Approach	5
2. OPTICAL FLOW	7
2.1 Overview	7
2.2 Optical Flow in Fly	10
2.3 Classification of Optical Flow	12
2.4 Optical Flow in Robotics and Computer Vision	13
3. THE FLY'S VISUAL SYSTEM	14
3.1 Overview	14
3.2 Compound Eyes	15
3.3 Organization of Fly's Motion Vision Pathways	17
3.4 Elementary Motion Detectors (EMDs)	18
3.5 Ambiguity of Local Motion Information	21
3.6 Directionally Selective Lobula Plate Tangential Cells (LPTCs)	22
3.7 Output LPTCs	23
3.8 Optical Flow Parameters Estimated by HS and VS Cells	26

4. IMPLEMENTATION: AN AGENT MODEL	29
4.1 Overview	29
4.2 Model of the Agent and the Environment	29
4.3 Computational Model of the Fly Visual System	31
4.3.1 Input Processing	31
4.3.2 Sensory Primitives	33
4.3.3 Learning Algorithm	38
5. EXPERIMENTS AND RESULTS	41
5.1 Experimental Setup	41
5.2 Results	42
5.2.1 Scenario 1	44
5.2.2 Scenario 2	51
5.3 Analysis	58
6. DISCUSSION	60
7. CONCLUSION	61
REFERENCES	62

LIST OF FIGURES

FIGURE	Page	
1.1	Decoding neural spikes. Decoding the meaning of neural spikes from (a) an external observer’s perspective where both the spikes S and the environmental stimulus I are available to the observer, compared to (b) an internal observer’s perspective (the brain itself) where only the spikes S are available. Adapted from [1].	2
1.2	The flynculus. The same problem of decoding neural spikes shown in the context of the fly (cartoon by M. F. Land, text by S. B. Laughlin [“The Flynculus: The little fly sitting in the fly’s brain, trying to fly the fly”]; as cited in [2], p14).	3
2.1	Optical flow detection. a) Image of visual features formed by projection of light from these features on the retina. When there is relative motion between the visual feature and the eye, a change in projected image on the retina is observed. This changing pattern of light on the retina due to relative motion between the observer and the visual scene is called optical flow. b) Shows the effect of change in light patterns at the pixel level. It also shows that by tracing the movement of a given pixel across different frames, the optical flow direction and velocity can be plotted as vectors. Adapted from [3]	8
2.2	Observed optical flows (Ideal). Ideal Optical flow fields generated by rotational and translational self motion are represented in vector notation. Observed optical flows induced by rotation (yaw, pitch, roll) and translation (forward and backward only) are shown.	9
2.3	Self motion generated optical flow. (A) Self-motion can be represented as components of translation and rotation along cardinal body axes, respectively. (B) Shows a small region in the compound eye of the fly. (f=frontal, c=caudal, d=dorsal, v=ventral). (C) Optical flow fields generated by upward translation motion as represented on the cylindrical projection (right) and surface of a unit sphere (left). (D) Shows clockwise roll rotation. Adapted from [4]	11
3.1	Cross section of the fly’s ommatidium. Adapted from [5].	16

3.2	Cross section of compound eye of the fly. Adapted from [6]	17
3.3	Horizontal section of the fly’s visual system. The illustration shows the retina (R) and the neuropils. The lamina (L) is connected to the medulla (M) via the external optic chiasm (CHE). Medulla is connected through internal optic chiasm (CHI) to lobula complex which consists of the anterior lobula (LO) and the posterior lobula plate (LP).The whole system is retinotopically organized meaning information from nearby points in the visual field are processed in adjacent columns. Adapted from [7]	18
3.4	Elementary motion detectors (EMD). EMDs detect local directional information by spatio-temporal correlation of light intensity of neighboring location on the retina. (A) Half-Detector produces output by multiplying delayed input from point 1 to input from point 2, depending on the direction of point’s alignment (preferred direction), the output can be positive or (B) negative for null direction. (C) and (D) are two mirror-symmetric half-detectors, giving positive response to motion in preferred direction while negative for null direction motion. Adapted from [8]	20
3.5	Ambiguity generated by local optical flow fields. Highlighted region in both images - (a) Left-Right Yaw (RYLR) and (b) Clockwise Roll (RRCL), do not give any idea of global optical flow pattern.	22
3.6	Distribution of dendrites of the fly’s right eye lobula plate tangential cells (LPTCs). HS dendrites (top row) span across the frontal(f)-caudal(c) direction responding best to front-back motion. VS dendrites (bottom row) span across the dorsal(d)-ventral(v) direction responding best to up-down motion.The sensitivity of the neuron’s response across the field of view depends on the density of the dendrites. Adapted from [4].	24
3.7	Preferred rotational axes of matched filters for VS cells of the right eye. The upper panel shows the preferred axes and sense of rotation of each VS cell (VS1-VS10) averaged over several individual flies. The diameter of circular arrows is directly proportional to standard deviation of axis of rotation across individual flies. Lower panel shows mean axis of rotation with VS4-VS10 shown to approximately lie on a line with 12° slope. Adapted from [4].	27

4.1	Model of the agent. The figure shows the problem of decoding internal sensory state from different point of views. (a) External Observer has access to both input I and state s , while (b) Internal Observer can only see internal states s . Here f are sensory filters that generate sensory state commensurate to external input. Adapted from [1].	30
4.2	Weight distribution of mVS and mHS neurons. The 2-D Gaussian plots show weight distribution of mVS and mHS neurons across the field based on dendrite distribution and density pattern (Fig. 3.6).	35
4.3	Reward table $R(s, a)$ with ideal reward values.	40
5.1	Input images. (a) Synthetic image with high contrast and low texture. (b) Indoor image with low contrast and high texture. (c) Natural image with low contrast and high texture. (d) Natural image with high contrast and high texture.	42
5.2	Initial and expected ideal reward table $R(s, a)$. (a) Initial randomly generated reward table. (b) Expected Ideal final reward table (Identity Matrix).	44
5.3	Final reward table $R(s, a)$ in Scenario 1. Final reward table obtained after training for Scenario 1 on the input images (in the same order as in Fig. 5.1).	45
5.4	Error plot in Scenario 1. Shows progression of error over the training iterations for Scenario 1. The error is calculated based on Equation 5.1. The error plots are generated for the four input images (in the same order as in Fig. 5.1).	46
5.5	Progress plot for input image Fig. 5.1(a) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(a) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	47

5.6	Progress plot for input image Fig. 5.1(b) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(b) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	48
5.7	Progress plot for input image Fig. 5.1(c) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(c) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	49
5.8	Progress plot for input image Fig. 5.1(d) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(d) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	50
5.9	Final reward table $R(s, a)$ in Scenario 2. Final reward table obtained after training for Scenario 2 on the input images (in the same order as in Fig. 5.1).	52
5.10	Error plot in Scenario 2. Shows progression of error over the training iterations for Scenario 2. The error is calculated based on Equation 5.1. The error plots are generated for the four input images (in the same order as in Fig. 5.1).	53

5.11	Progress plot for input image Fig. 5.1(a) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(a) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	54
5.12	Progress plot for input image Fig. 5.1(b) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(b) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	55
5.13	Progress plot for input image Fig. 5.1(c) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(c) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	56
5.14	Progress plot for input image Fig. 5.1(d) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(d) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.	57

LIST OF TABLES

TABLE		Page
4.1	Parameter values of modeled HS cells (mHS).	34
4.2	Parameter values of modeled VS cells (mVS). Some parameters that are common among all mVS are: Elevation Range= $[-45^\circ, 45^\circ]$, $\sigma_\phi = 24^\circ$, $\sigma_{\theta+} = 24^\circ$ and $\sigma_{\theta-} = 37.5^\circ$	36
4.3	Matched filter parameters of modeled VS cells (mVS). The table gives the center of axes of rotation along which mVS cells are matched to respond.	37

1. INTRODUCTION

1.1 Overview

The fly visual system, although tiny when compared to the mammalian visual system, can still perform highly sophisticated functions to help the animal navigate through the environment (see [9] for a general review of insects' spatial understanding). An example of such a function is the detection of optical flow. The behavior and anatomy of the optical flow system in the fly have been extensively studied [10] [11] [4] [12] and modeled computationally [12] [13] [14]. Perceiving and responding to the optical flow is critical to the animal's survival and success in life, since based on this they perform rapid and complex maneuvers such as stabilization and pursuit. Higher level neurons in the fly's visual system, the Lobula Plate Tangential Cells (LPTCs), respond to different types of optical flow due to rotation and translation of the animal, by pooling information from elementary motion detectors (EMDs) in the lower level [15] [16] [17]. In this sense, neuronal responses (spikes) from these optical flow detectors in the fly carry highly encoded signals.

In this thesis, I investigate how such highly encoded optical flow signals can be interpreted and utilized within the fly's brain, while solely operating on the internal spike patterns in its own brain (Fig. 1.1). In some sense, the question is like the cartoon shown in Fig. 1.2: how can the fly, sitting inside its brain, figure out how to interpret the spikes and generate relevant action (the "Flynculus", a fly version of the homunculus). This is basically a problem of "grounding" [18], i.e., trying to figure out the meaning of internal representations in the brain. Strangely, the grounding problem seems trivial from an external observer's point of view, with complete access to the stimulus and the internal spikes (Fig. 1.1a), while it seems intractable from

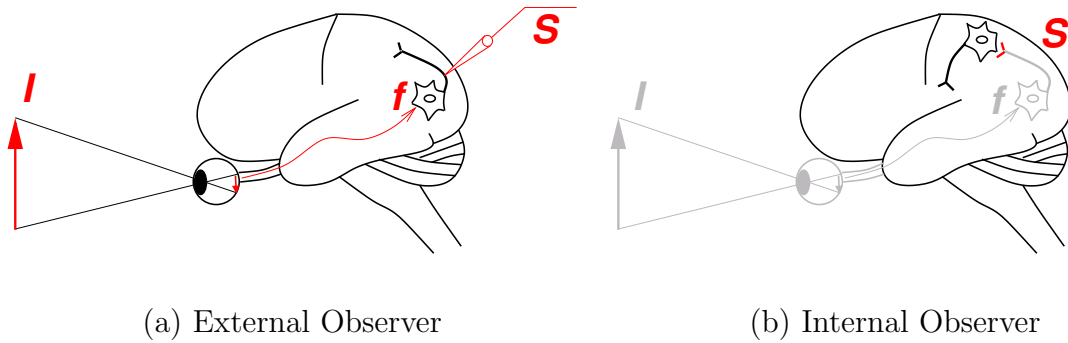


Figure 1.1: Decoding neural spikes. Decoding the meaning of neural spikes from (a) an external observer’s perspective where both the spikes S and the environmental stimulus I are available to the observer, compared to (b) an internal observer’s perspective (the brain itself) where only the spikes S are available. Adapted from [1].

an internal observer’s view (the brain’s native view; Fig. 1.1b).

In previous work on grounding visual cortical orientation detector responses, it has been shown that action is critical in making the internal observer (Fig. 1.1b) case feasible [1][19]. With a computational model of the optical flow detectors in the fly, I show that action (or coordinated motor output) is the only way that the fly can learn the meaning of its internal spikes and generate meaningful, relevant behavior, all just based on internal spike patterns.

1.2 Background

Optical flow is the change in structured light in an image due to relative motion between the scene and the observer. Optical flow is very important in understanding motion in various animals. It has been shown that optical flow has a significant role in perception of motion, distinguishing objects, depth perception and control of locomotion. Optical flow has also been used extensively by robotics researchers in object detection, navigation, depth perception, etc. Optical flow is discussed in

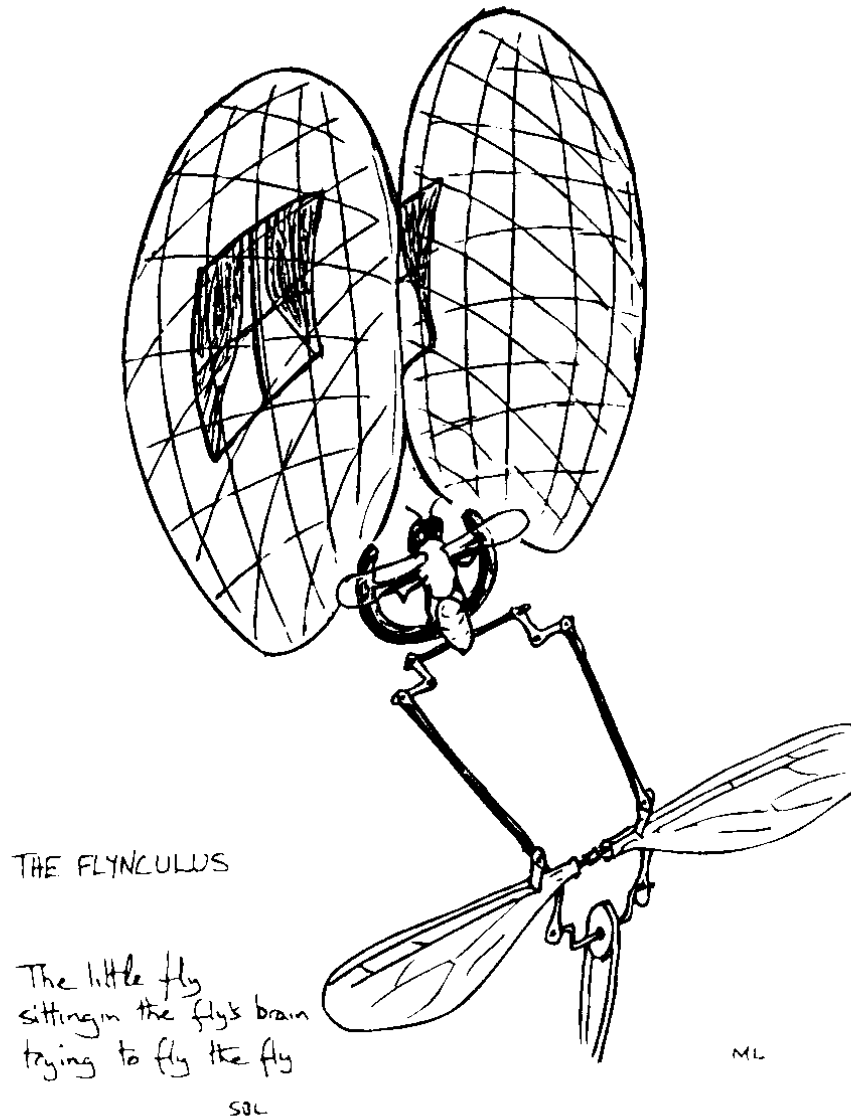


Figure 1.2: The flynculus. The same problem of decoding neural spikes shown in the context of the fly (cartoon by M. F. Land, text by S. B. Laughlin [“The Flynculus: The little fly sitting in the fly’s brain, trying to fly the fly”]; as cited in [2], p14).

detail in Section 2.

Flying insects have been studied extensively in the past decades. The structure of their brain; structural and functional models of neurons involved in sensing; processing; and giving motor response in different modalities have been studied extensively. There are now mathematical, structural, and functional models available for many neurons involved in important functions of the fly. It is easier to do this analysis for the fly brain because of its small and relatively simple structure. Each functionality only uses a limited number of neurons so it becomes easier to study their functionality. The fly visual system is also of significant interest as far as this research is concerned as flies are involved in rapid motion activities like pursuit of prey, gaze stabilization, etc. These activities require a robust mechanism that can deal with different kinds of optical flows.

The fly visual system consists of three visual neuropiles and the retina, which is connected to a hexagonal eye lattice of the compound eye. The retina has local motion detectors called Elementary Motion Detectors (EMDs). These EMDs respond to motion in local regions of the field of view and are directionally selective. Signals generated by EMDs are then integrated by the Lobula Plate Tangential Cells (LPTCs). These are integrator neurons that selectively integrate input from EMDs to generate a response that is commensurate to the global motion as perceived across the field of view. There are about 60 different LPTCs, responding to different aspect of motion. However, Horizontal System (HS) and Vertical System (VS) neurons are the most significant ones. HS cells respond to horizontal motion while VS cells respond to vertical motion. The response of these cells depends on velocity of motion and contrast of the scene. These neurons connect to output neuropiles that control motion. Thus, study and modelling of these neurons is directly relevant to this research. The fly's visual system is described in detail in Section 3.

1.3 Approach

The approach that I have followed is to study the literature for models of these neurons in the fly's visual system and perform experiments by implementing these models with a learning algorithm on top of it to derive a sensorimotor mapping. The existence of such a mapping justifies the hypothesis that agents can learn the meaning of sensory input in terms of its actions, which is the grounding of the optical flow. The objective is to show these results for both artificial scenarios and natural scenarios.

I used images with different types of background ranging from indoor scene to outdoor natural scene. These images were used to simulate a scene that a fly-like robotic agent observes during the training phase. These images were moved to generate relative motion between the agent and the scene, causing the agent to observe different types of optical flows like roll, pitch, yaw, and zoom. These formed the optical flows to model those generated by self-motion of a fly-like agent. I used standard optical flow generation methods such as Lucas-Kanade method and Horn-Schunck. The next step is to build a model of visual processing system inspired by the fly's visual system, which generates different firing outputs for different types of optical flows. These sensory states are generated by models of HS and VS neurons. These firing states act as an input for the learning algorithm. The learning algorithm is based on reinforcement learning and performs different actions to optimize the reward. The reward is generated based on the invariance criterion, meaning that the actions that do not change the sensory state generate high reward. This technique creates a sensory-motor mapping from a particular optical flow state to a particular action. This mapping serves as the basis of grounding. The more accurate the mapping, the better the grounding can be. The approach and a model of a robotic

fly-like agent, capable of autonomously grounding optical flow, is presented in detail in Section 4.

In Section 5, I describe various experiments conducted with different images, both natural and synthetic. I also conducted various experiments by varying parameters of the agent model and learning algorithm. Results with varying conditions are shown giving good sensory-motor mapping in some cases while worse in others. Generally, good results were obtained for images simulating natural scenarios that exhibit high contrast scenes with dense texture.

2. OPTICAL FLOW

2.1 Overview

Optical flow is the change in structured light in an image due to relative motion between the scene and the observer. Optical flow is very important in understanding motion in various animals. It has a significant role in estimation of the self-motion, distinguishing objects, depth perception and control of the locomotion. The study of optical flow has placed great emphasis on estimating the 3-D nature and structure of scenes and analyzing motion between objects and observers. Optical flow has many applications in the field of biology, computer vision, robotics, and image compression.

In the biological contexts, optical flow is a change in light patterns that gives the impression of movement of visual imagery that is projected onto the retina. This is illustrated in the Fig. 2.1 in detail. Directional motion is processed in all visually oriented animals including human beings (see [20] [21]). Information gained from visually processed directional motion gives a good understanding of environment and self-orientation to the animal. For example, a human being with only one eye can still manage to identify objects by distinguishing background from foreground. This is achieved by moving the head from side to side to generate an optical flow that is then used to understand these features of the environment. Optical flow is also useful for extracting depth information.

Gibson [22] first introduced the notion that optical flow is a rich source of information related to self motion and distance. He showed that optical flow plays an important role in affordance perception which is the ability of animals to discern the possibilities for actions within the environment. Later, the formal method of describing optical flow in vector notation format was proposed (see [23] [24]). This

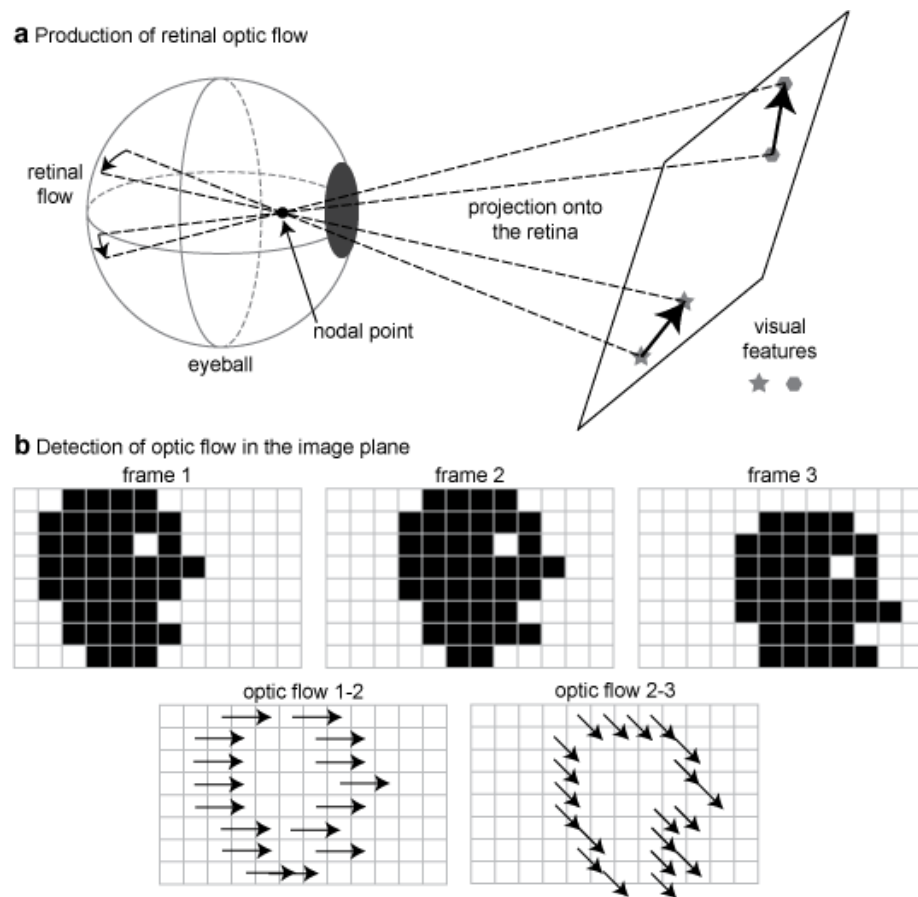


Figure 2.1: Optical flow detection. a) Image of visual features formed by projection of light from these features on the retina. When there is relative motion between the visual feature and the eye, a change in projected image on the retina is observed. This changing pattern of light on the retina due to relative motion between the observer and the visual scene is called optical flow. b) Shows the effect of change in light patterns at the pixel level. It also shows that by tracing the movement of a given pixel across different frames, the optical flow direction and velocity can be plotted as vectors. Adapted from [3]

format of representing optical flow as shown in Fig. 2.2 gives a good idea of direction and velocity of moving pixels. It is very useful for analyzing optical flow at local levels as well as inferring global flow patterns.

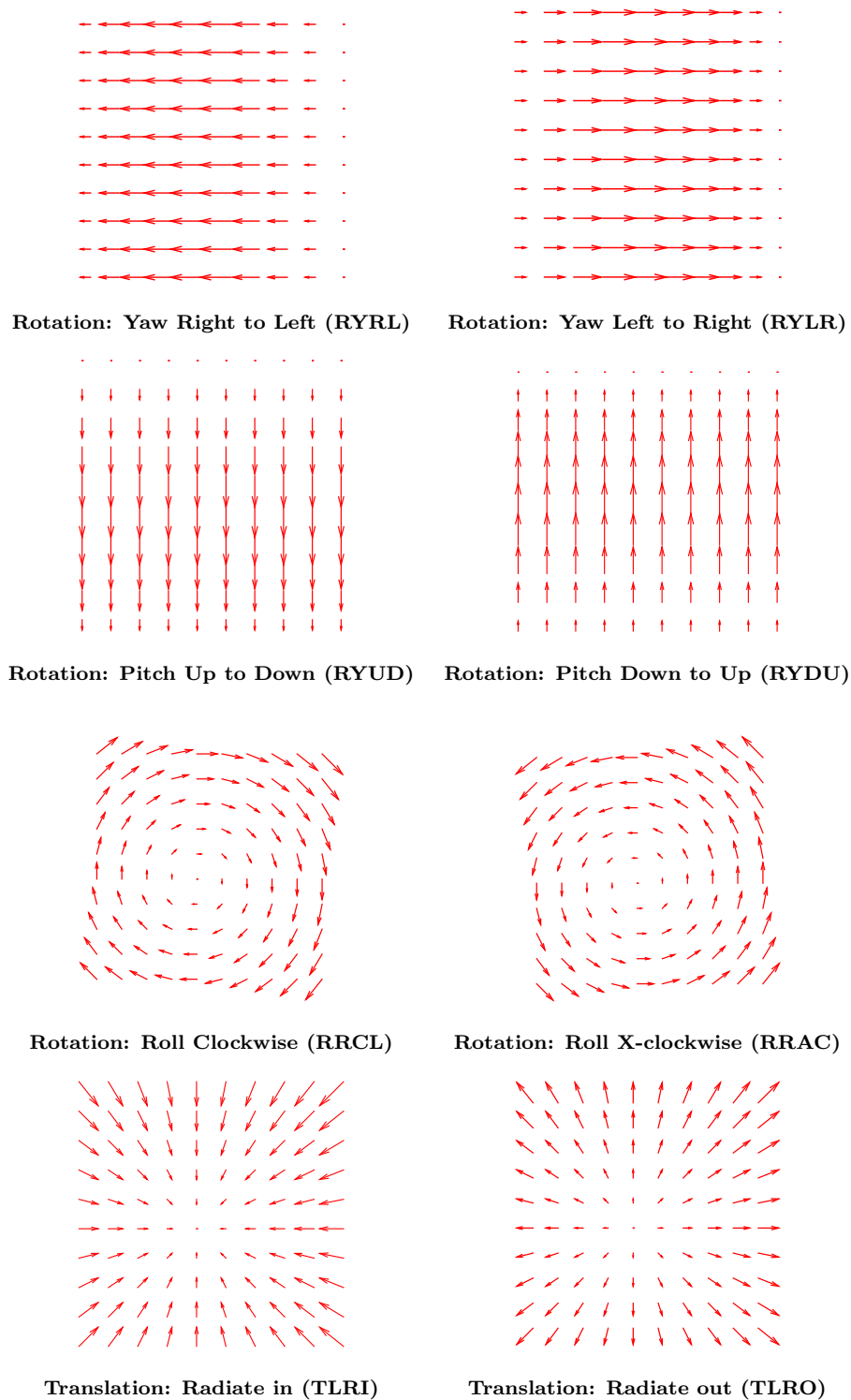


Figure 2.2: Observed optical flows (Ideal). Ideal Optical flow fields generated by rotational and translational self motion are represented in vector notation. Observed optical flows induced by rotation (yaw, pitch, roll) and translation (forward and backward only) are shown.

2.2 Optical Flow in Fly

The fly has a robust visual system that senses and processes visual motion information. Understanding directional motion information such as optical flow is very important for performing actions like gaze stabilization, pursuing prey, etc. The fly performs these behaviorally meaningful actions in an exceedingly efficient manner. Therefore, the fly is a good example to study characteristics of optical flow and its processing mechanism as discussed in greater detail in Section 3.

Consider that a fly as shown in Fig. 2.3 is a rigid object that is capable of translating in 3-D space and rotating along 3 axes leading to yaw, pitch, or roll motion. When the fly performs these actions, it senses optical flow through its visual system (described in detail in Section 3) and generates a global map of flow fields. These flow fields can be represented using vector notation as shown in Fig. 2.3. In this representation, two angles become very important for representing any point in the field: the azimuth and elevation angles, respectively. An azimuth and elevation of 0 represents the point is right in front of the fly. Azimuth of >0 and <0 represent position of a point in right and left visual hemispheres, respectively. Elevation of >0 and <0 represent position above and below the horizon, respectively.

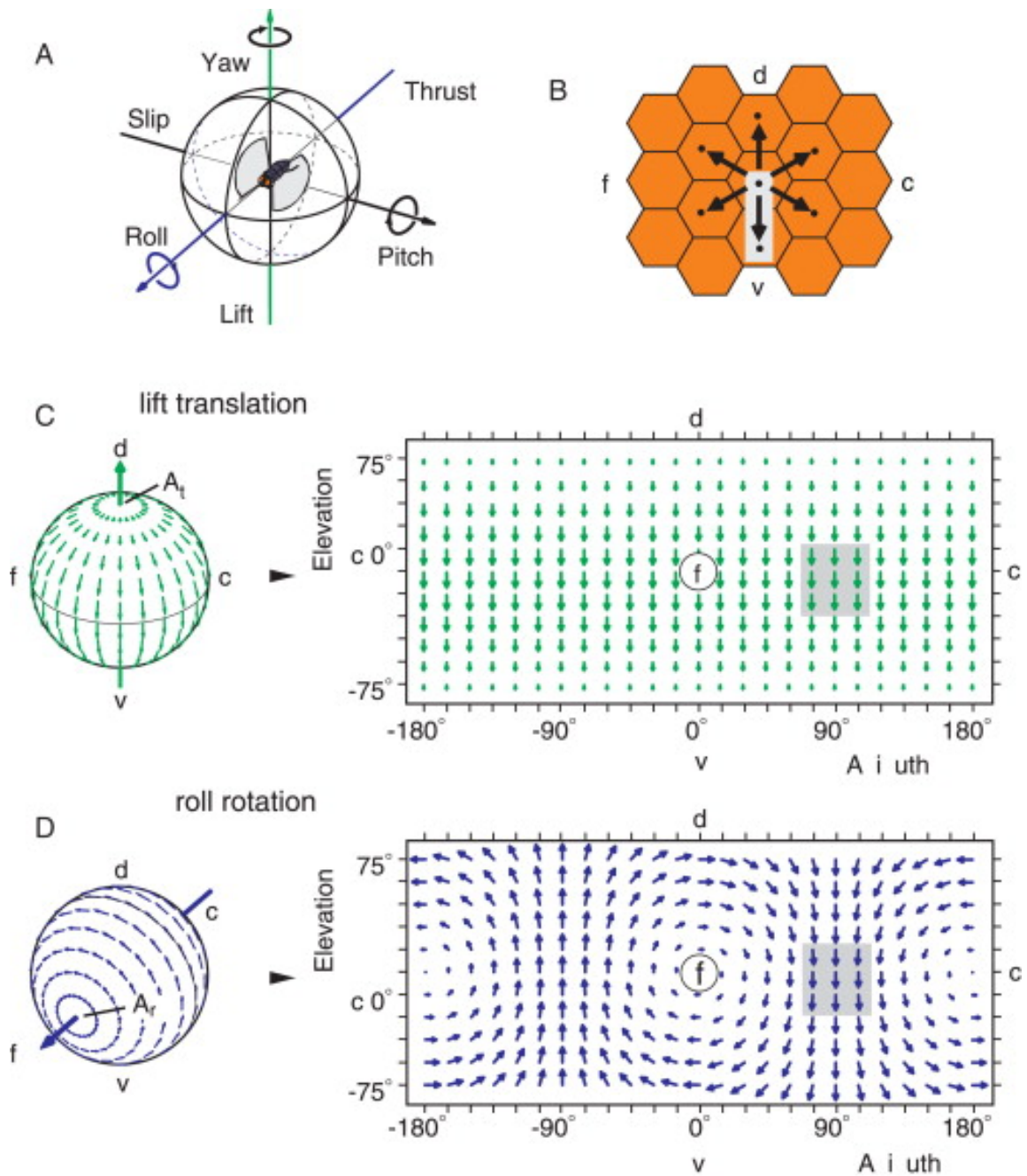


Figure 2.3: Self motion generated optical flow. (A) Self-motion can be represented as components of translation and rotation along cardinal body axes, respectively. (B) Shows a small region in the compound eye of the fly. (f=frontal, c=caudal, d=dorsal, v=ventral). (C) Optical flow fields generated by upward translation motion as represented on the cylindrical projection (right) and surface of a unit sphere (left). (D) Shows clockwise roll rotation. Adapted from [4]

2.3 Classification of Optical Flow

The orientation and length of the arrows in the optical flow plots indicate the direction and magnitude of optical flow velocity at a given point. Looking at this optical flow field pattern, it is possible to identify the type of relative motion that generated it. For example, the optical flow patterns where an entire flow field follows the same pattern is called global optical flow pattern. Such a pattern is generated by the fly's self motion. Another type of optical flow is called local optical flow pattern which is generated by external object's motion such as a moving prey in the environment.

Global optical flow patterns can be classified as translational, rotational, or mixed optical flows. These are generated by translational motion, rotational motion, or some combination of both between the scene and the observer. Pure rotational and translational optical flows have a common feature that the axis of rotation and direction of translation are defined by points in the visual field where no relative motion takes place. These are called flow field singularities. Starting from these points, relative motion gradually increases to be maximum at the equator. However, real life maneuvers are a combination of rotational and translational optical flows, i.e. mixed optical flows, and do not have singularities coinciding with either direction of translation or axis of rotation.

Apart from these differences, two key structural difference separate translational and rotational optical flow fields (see [4]):

1. Rotational fields have local velocity vectors aligned along parallel circles that are centered on the axis of rotation, while translational fields have velocity vectors aligned along circles connecting two flow field singularities.
2. The magnitude of local vectors depends on the distance to the objects in the

environment in translational optical flow fields. The magnitude of local vectors is independent of distance for rotational optical flow fields. This means that translational optical flow provides relative distance information which the rotational optical flow doesn't. This information is important for behaviors like obstacle avoidance.

2.4 Optical Flow in Robotics and Computer Vision

Optical flow has also been used extensively by robotics researchers in object detection and tracking, image dominant plane extraction, robot navigation, movement detection, and visual odometry. It is also very useful in video compression and motion estimation research. It is used in applications such as inferring the motion of the observer and objects in the scene, determining the structure of objects, etc.

In computer vision, many methods have been proposed to determine optical flow from a sequence of images or video. Representative methods are listed below:

1. **Differential Methods** like Lucas-Kanade Method [25] and Horn-Schunck Method [26]. These methods use partial derivatives of the image signal and additional conditions like brightness constancy constraint to generate optical flow.
2. **Phase Correlation Method.** This method uses a fast frequency-domain approach that estimates the relative translative offset between the two images.
3. **Block-based Method.** This method minimizes the sum of the squared difference or maximizes normalized cross-correlation between two images.

3. THE FLY'S VISUAL SYSTEM

3.1 Overview

The fly has a very tiny brain. Yet it is capable of exhibiting seemingly complex behaviors such as escape, landing, controlling altitude and thrust, pursuing mates and prey, flight stabilization, etc. It is almost impossible to exhibit such behavior without using long range sensor information as is generated by visual sensors. Visual sensory system, unlike mechanoreceptor-based system which requires the action of local physical forces, can be used for both long-range identification of targets and solve short range feature extraction. Another advantage is that information about different types of visual qualities can be extracted simultaneously using parallel pathways that process these qualities separately. For example, one visual pathway would process the motion information, while simultaneously another pathway color information.

The fly has a very simple yet robust mechanism for sensing and processing visual information. Due to its relatively simple neural mechanism, it is relatively easier to study and model the fly's visual system than that of the vertebrate. Thus, it has been studied extensively in the fields of zoology and bioengineering. One of the most interesting aspect of the fly's visual system is its ability to process complex motion information allowing it to perform rapid rotational and translational actions. The ability to process directional motion exists in many living organisms including humans. However, the simplicity of the fly's visual processing system and its ability to still exhibit most complex motion responses makes it ideal for study in this research. The fly's visual system consists of spatially distributed motion detectors, spanning the visual field, that sense the motion in the surrounding area. The information col-

lected by these detectors is integrated by specialized neurons that generate response commensurate to the global motion information. Finally, the processed visual information along with information from other sensory modalities are presented to the motor neurons that control the muscles that eventually produce the behaviours.

In this Section, I will give an overview of the overall architecture of the fly's visual system. Though the study of the fly's visual system is vast and consists of lot of literature, I will emphasize only details that are of direct relevance to this research problem. Some of the background information related to fly's visual system that has been used in this section is taken from literature reviews (see [4] [27]).

3.2 Compound Eyes

The fly has 2 compound eyes in the anterior region of its body. Each eye consists of tens of thousands of sensory units called ommatidia. The exact count of ommatidia in each compound eyes varies across different species of fly. An ommatidium consists of a group of photoreceptor cells that are surrounded by pigment cells and other support cells (see Fig. 3.1). The ommatidium is covered from outside by a transparent cornea. This allows the light to pass through without distortion while being focused on the photoreceptors. Pseudocone is present below the cornea that helps to further focus the light. Ommatidia are hexagonal in shape with diameter largest at the surface while tapering toward the inner end.

The Compound eye is made up of a lattice of these hexagonal ommatidia (see Fig. 3.2) that are instrumental in providing a wide-angle panoramic view of the surround to the fly. This wide field of view gives the fly its ability to locate the targets without necessarily scanning the surrounding by rotating its head. Thus, the compound eye is the interface between the fly's neural system and the outside environment and houses photoreceptors that generate electrochemical signals that are processed by

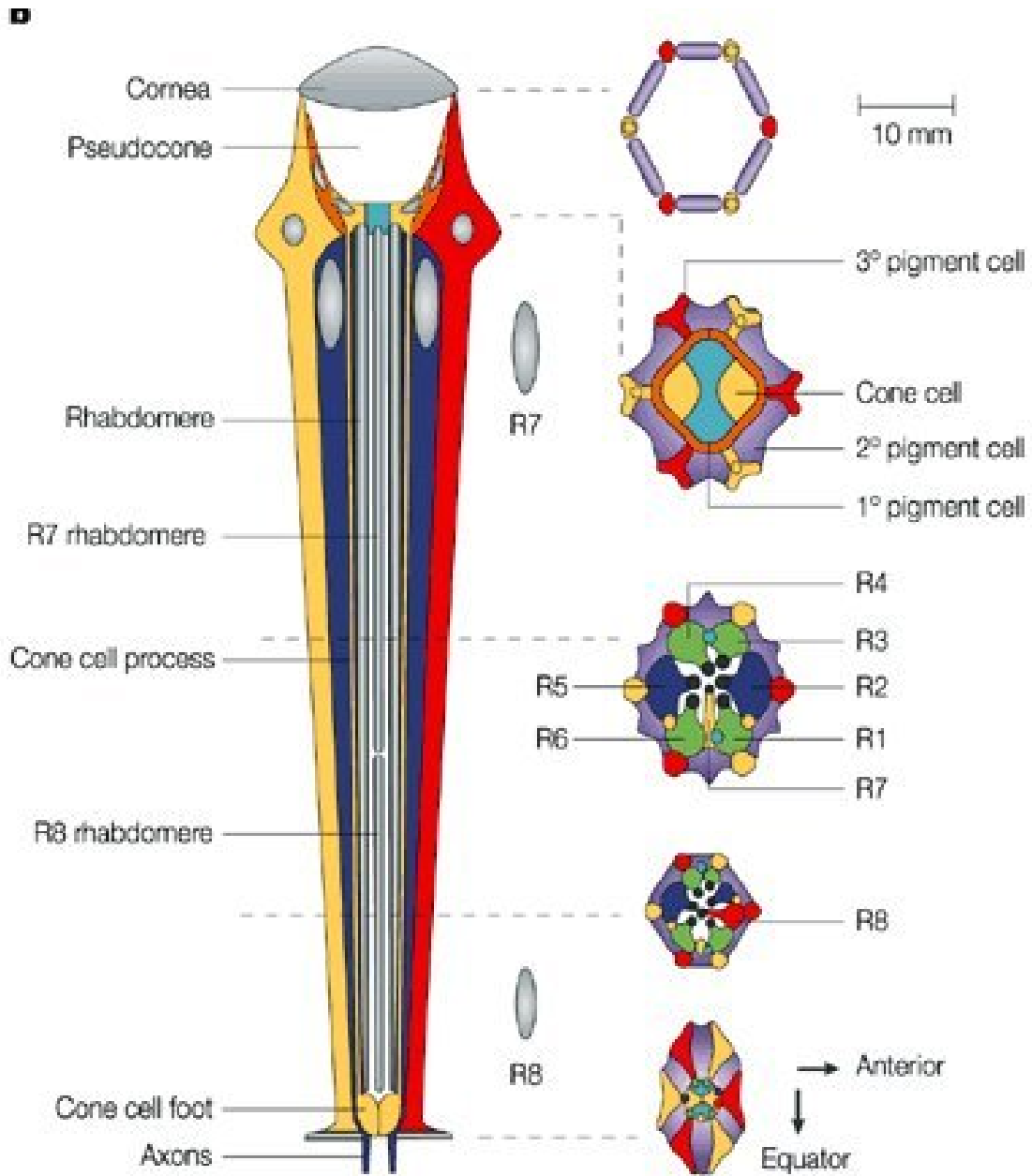


Figure 3.1: Cross section of the fly's ommatidium. Adapted from [5].

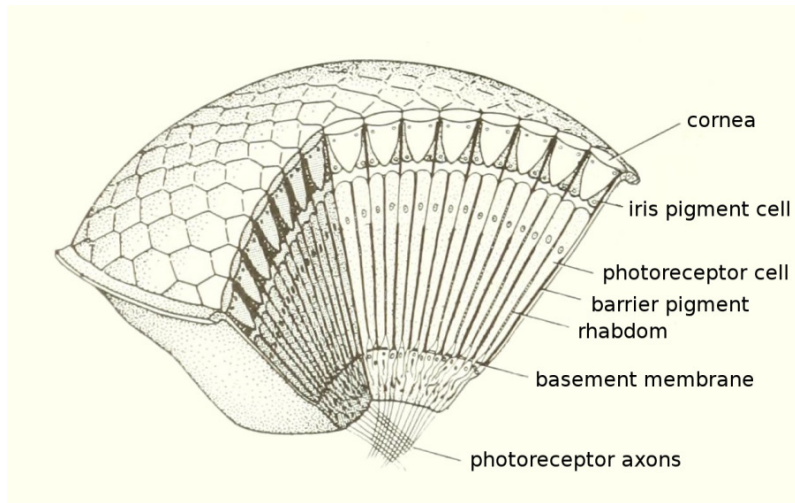


Figure 3.2: Cross section of compound eye of the fly. Adapted from [6]

subsequent neurons.

3.3 Organization of Fly's Motion Vision Pathways

The ability of the fly's visual system to efficiently understand motion and its direct relation to this research thesis is a compelling motivation to look at the visual pathways responsible for processing motion information in greater detail. The organization of visual motion pathways as seen most commonly across various fly species is described below.

The fly visual system consists of three visual neuropiles and the retina (see Fig. 3.3). The retina is connected to the hexagonal lattice of the compound eye and has photoreceptors that respond to light stimuli. The lamina, the first neuropile, receives input from the retina's photoreceptors. This visual information is then passed on to the medulla, the second neuropile via an external chiasm. From the medulla, the connections are made to the lobula complex, which consists of the anterior lobula and the posterior lobula plate. There exists an internal chiasm between the medulla and the lobula plate. The lobula plate is the major center for integration of visual

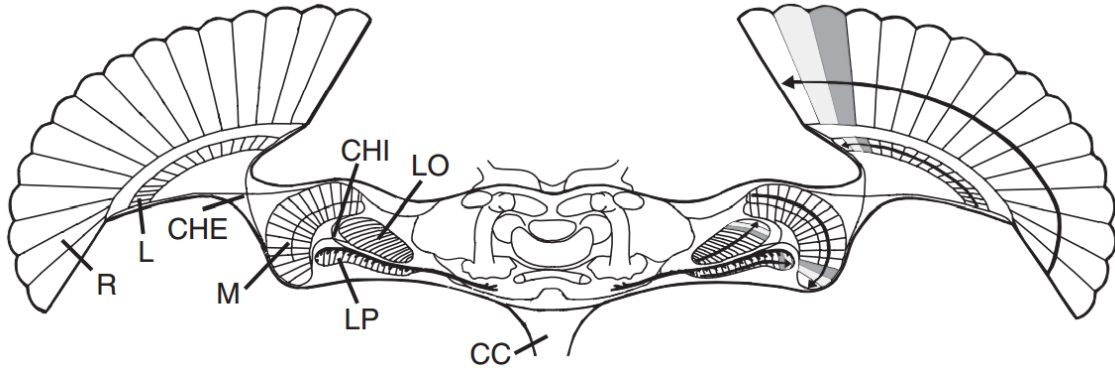


Figure 3.3: Horizontal section of the fly’s visual system. The illustration shows the retina (R) and the neuropils. The lamina (L) is connected to the medulla (M) via the external optic chiasm (CHE). Medulla is connected through internal optic chiasm (CHI) to lobula complex which consists of the anterior lobula (LO) and the posterior lobula plate (LP).The whole system is retinotopically organized meaning information from nearby points in the visual field are processed in adjacent columns. Adapted from [7]

information collected by the retina. The entire system is retinotopically organized and the processing of visual information takes place in separate columns. This means that the information derived from adjacent points in the visual field is processed in adjacent columns throughout the system, right from the retina to the lobula plate. The two chiasms invert the image twice along the pathway.

3.4 Elementary Motion Detectors (EMDs)

Any system that aspires to exploit global optical flow information must first be able to collect spatially distributed local directional motion information spanning the complete field of view. The fly achieves this through an array of motion detectors called Elementary Motion Detectors (EMDs) spread over the entire field of view. EMD is the theoretical model of neural mechanism present between lamina and medulla that process the changes in the light intensity as sensed by photoreceptor cells to generate local directional motion information. This motion information is

based on changes taking place in the surroundings.

Many behavioral experiments have been performed to derive a phenomenological model (see [28]) of the Elementary Motion Detector (EMD). Many attempts have been made (see [16] [8] [29]) to establish functional and physiological models. Various motion detector schemas based on experimental evidence obtained in different animal systems or from computer vision are discussed (see [30]).

Basically every EMD model satisfies the following necessary and sufficient condition required for the analysis of directional motion:

1. Two signals from spatially adjacent locations, obtained from adjacent ommatidia.
2. Processing the two signals in an asymmetric manner: delay one signal while keeping other unchanged.
3. A non-linear operation on the two signals: multiply delayed and non-delayed signal.

Consider the model of EMD as shown in Fig. 3.4. Here, the EMD generates a positive output in response to motion in its preferred direction while no response in the opposite direction called the null direction. Such a detector is called a half-detector as it responds to motion in only one direction. A fully directionally selective EMD is formed by adding output of two mirror symmetric EMDs with opposite signs. This EMD model is also called the correlation type motion detector, which gives positive response to motion in the preferred direction while negative response in the opposite direction.

It is clear that the EMD responds only in direction in which they are arranged. Typically, EMDs are arranged in various orientations ranging from horizontal or

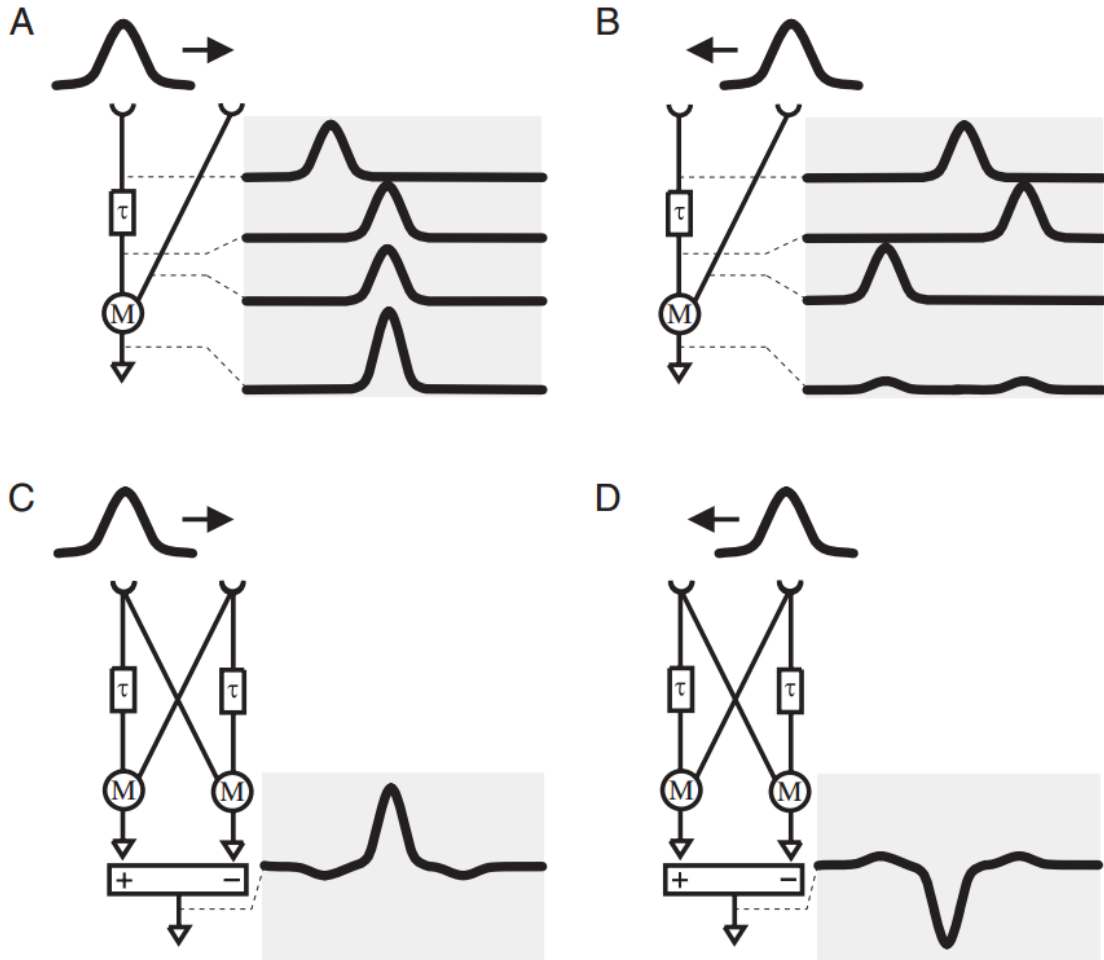


Figure 3.4: Elementary motion detectors (EMD). EMDs detect local directional information by spatio-temporal correlation of light intensity of neighboring location on the retina. (A) Half-Detector produces output by multiplying delayed input from point 1 to input from point 2, depending on the direction of point's alignment (preferred direction), the output can be positive or (B) negative for null direction. (C) and (D) are two mirror-symmetric half-detectors, giving positive response to motion in preferred direction while negative for null direction motion. Adapted from [8]

vertical or at an angle to either axes. It is the property of the integration neurons as to what global orientation of EMDs they take their input from as discussed in Section 3.7.

The EMDs have the property that their response depends directly on the velocity and direction of visual motion. Its response is also affected by contrast and spatial frequency content of the visual scene. The biological mechanism of the EMDs in certain species is hypothesized to be similar to a gradient detector, where output depends upon velocity and direction. The difference is that the gradient model gives output linear in the velocity domain whereas the EMD correlation type motion detector model, which is the accepted model, gives response that vary according to the spatial frequency properties of the scene. Many complex models of EMDs have been suggested that take into account gain control mechanism and properties of the peripheral visual system. I will not discuss these models in detail in this document.

3.5 Ambiguity of Local Motion Information

It is now clear that the fly uses an array of EMDs to detect local motion and correspondingly generate a graph of optical flow field information spread across the entire field of view. It however does not give any information about the global optical flow pattern. For example, consider the two optical flow patterns shown in Fig. 3.5. In the highlighted region, the local optical flow is aligned in horizontal direction from left to right in both patterns. However, it is impossible to make the judgment whether it is left-right yaw or clockwise roll optical flow by just looking at the optical flow in the highlighted section. In order to distinguish between optical flows, the fly must selectively pool the spatially distributed local optical flow information to generate a response that depends on the global optical flow pattern. This is achieved in the fly by a set of neurons called the Lobula Plate Tangential Cells (LPTC). The cells

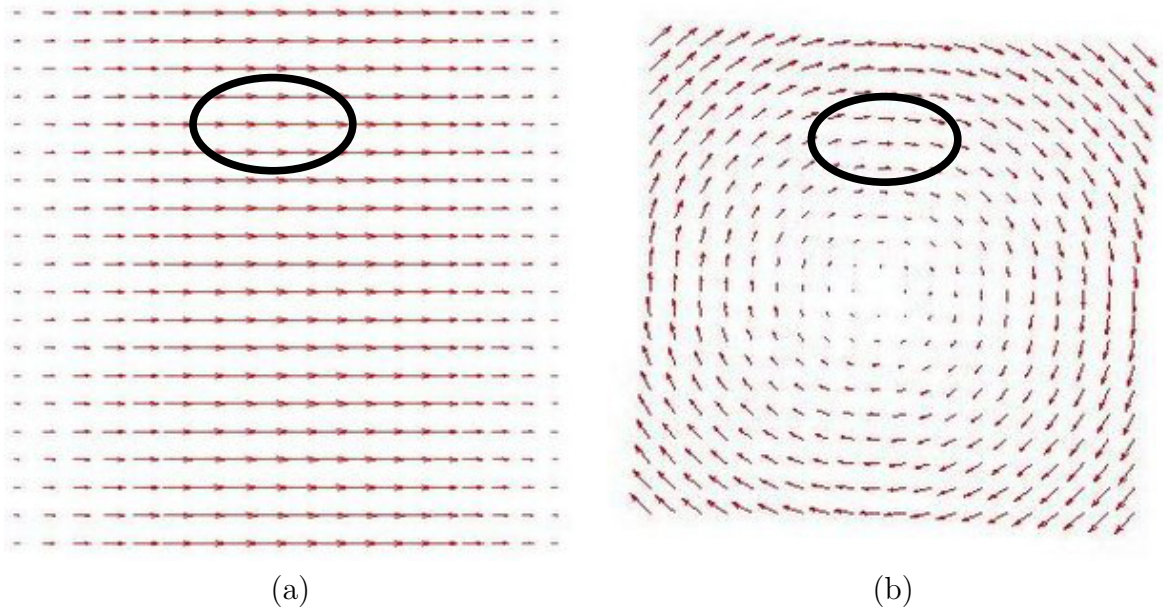


Figure 3.5: Ambiguity generated by local optical flow fields. Highlighted region in both images - (a) Left-Right Yaw (RYLR) and (b) Clockwise Roll (RRCL), do not give any idea of global optical flow pattern.

selectively integrate local motion information to generate a global picture. These cells are discussed in detail in Section 3.7.

3.6 Directionally Selective Lobula Plate Tangential Cells (LPTCs)

The next step in processing of visual motion information is the integration of local visual motion information to generate a response commensurate with the global visual motion pattern across the field of view. This is done by the Lobula plate tangential cells (LPTCs) in the fly's visual motion processing system. The lobula plate, the third neuropile, houses 60 visual interneurons, the lobula plate tangential cells (see [31]). LPTCs pool information from EMDs spread across the visual field through their dendrites that arborize in different directions according to the range of their receptive field and sensitivity distribution over that field. The output of these neurons is the weighted sum of local visual motion information generated by EMDs.

There are four functional groups of LPTCs that have been anatomically identified and electro-physiologically characterized (see [4]):

1. **Heterolateral LPTC:** These neurons receive input from extended parts of the visual field and they convey the information in the form of spike rate modulations to the lobula plate.
2. **Output LPTC:** They are the main neurons involved in collecting local visual motion directly from EMDs and respond to common rotational and translational optical flow patterns.
3. **Centrifugal LPTC (CH-cells):** They are involved in figure detection and receive input from various sources.
4. **Figure Detection (FD-cells):** They receive directionally sensitive information from EMDs and inhibitory signals from CH-cells. These cells specialize in detecting figure over small regions of field of view.

These different types of LPTCs are involved in a variety of tasks but the output LPTCs are of great importance for processing self-motion generated optical flows. I will describe output LPTCs in greater detail in the following sections.

3.7 Output LPTCs

Output LPTCs are the most significant form of interneurons responsible for processing rotational and translational optical flow information as generated by the fly's self motion. This makes the study of these LPTCs very important for this thesis. Output LPTC are of two distinct types:

1. Horizontal System cells (HS-cells) and
2. Vertical System Cells (VS-cells).

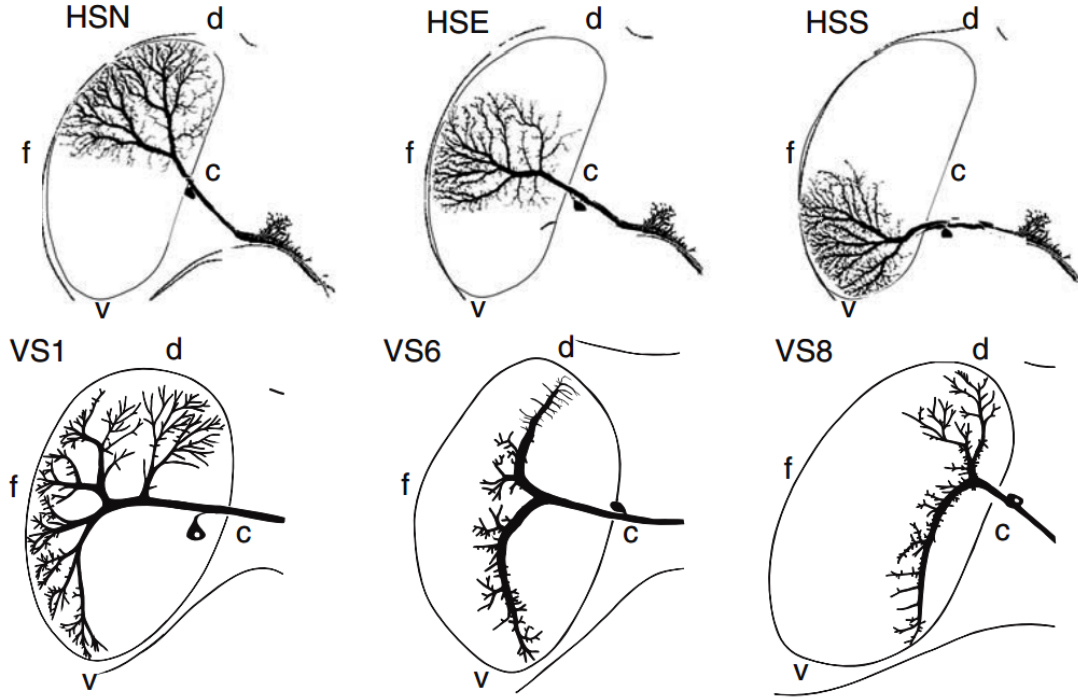


Figure 3.6: Distribution of dendrites of the fly’s right eye lobula plate tangential cells (LPTCs). HS dendrites (top row) span across the frontal(f)-caudal(c) direction responding best to front-back motion. VS dendrites (bottom row) span across the dorsal(d)-ventral(v) direction responding best to up-down motion. The sensitivity of the neuron’s response across the field of view depends on the density of the dendrites. Adapted from [4].

These names are used to categorize these cells based on their predominant directional preference for either vertical or horizontal motion.

The HS-cells respond to horizontal motion. Each eye has 3 corresponding HS-cells - north (HSN), equatorial (HSE) and south (HSS), that integrate visual information from dorsal, equatorial, and ventral parts of the ipsilateral visual field, respectively (see Fig. 3.6). The horizontal direction from front to side is the preferred direction of the HS-cells. As shown in Fig. 3.6, HS-cell dendrites arborize in horizontal frontal to caudal direction in the visual field. The sensitivity of the HS-cells to motion is

higher in the frontal region and reduces gradually towards the caudal direction. The sensitivity depends upon the density of dendrites in the given region. This means that local horizontal motion information received from EMDs in the frontal region is weighted higher than one in the caudal region. It is important to note that HS-cells mainly respond to inputs received from predominantly horizontally arranged EMDs while some weak signals are also received from heterolateral LPTCs.

The VS-cells respond predominantly to vertical motion. Each eye has 10 corresponding VS-cells - VS1 to VS10 that integrate visual information from the frontal visual field for VS1 to caudal visual field for VS10. Vertical direction from up to down is the preferred direction of VS-cells. As shown in Fig. 3.6, VS-cells dendrites are vertically oriented with dendrites arborizing in an up to down direction in the visual field. The sensitivity of the VS-cells to motion is higher in the dorsal region and reduces gradually in the ventral direction. This means that local vertical motion information received from EMDs in the dorsal region is weighted higher than in the ventral region. Its important to note here that VS-cells do not react to perfect vertical motion, they rather depend on a predominately vertical motion matched to a specific combination of pitch, roll, or yaw motion. This is discussed more in detail in Section 3.8.

Output LPTCs generate signals that are distinctly different from other LPTCs. These neurons get depolarized in response to motion in their preferred direction while they get hyper-polarized for motion in the opposite direction. Both HS and VS cells are equipped with excitatory and inhibitory receptors - ACh and GABA receptors, respectively. They receive input from two mirror-symmetric units of EMDs. The neurons depolarize when excitatory input is generated by motion in the preferred direction. This output is directly proportional to motion velocity. The sensitivity of these neurons to the EMD's motion information in their receptive fields vary

according to the density of dendrites in different regions in their receptive fields. Finally, the axon terminals of output LPTC cells connect to target neurons, that are mostly descending neurons that project to various motor neuropiles. Output LPTC terminals are located in an area where output fibers of neurons involved in other sensory systems are also present. Therefore, this is the place where information from various sensory modalities are pooled and presented to the motor neurons.

3.8 Optical Flow Parameters Estimated by HS and VS Cells

Originally output LPTCs have been named based on their predominant directional preference in their response. For instance, HS and VS cells have been named so as their response is predominantly to horizontal and vertical motion, respectively. This seems true when looking at the overall orientation of their dendrite branching. However, study has shown that none of the output LPTCs show strictly horizontal or vertical preference (see [32]). Rather, many experimental results support the hypothesis that each LPTC cell responds to a specific component of self-motion generated optical flow.

Studies have shown that the HSE cells react to optical flow generated by a yaw rotation around the vertical body axis. HSN-cell is tuned to an axis somewhere between yaw and nose-dive pitch. VS-cells are proposed to be matched to an optical flow field generated during a particular rotation around an axis roughly aligned with the horizontal plane. As shown in Fig. 3.7, VS1 is matched to optical flow suited to nose-up pitch. While VS10 is suited to nose-down pitch, other VS-cells are tuned to rotation along the intermediate horizontal axes, suited for some combination of roll and pitch. It is interesting to note the high degree of structure in the arrangement of axes of rotation with respect to which VS-cells are tuned to respond to. Axes of eight out of ten VS-cells lie on a line with a slope of 12° . This means that VS6 responds

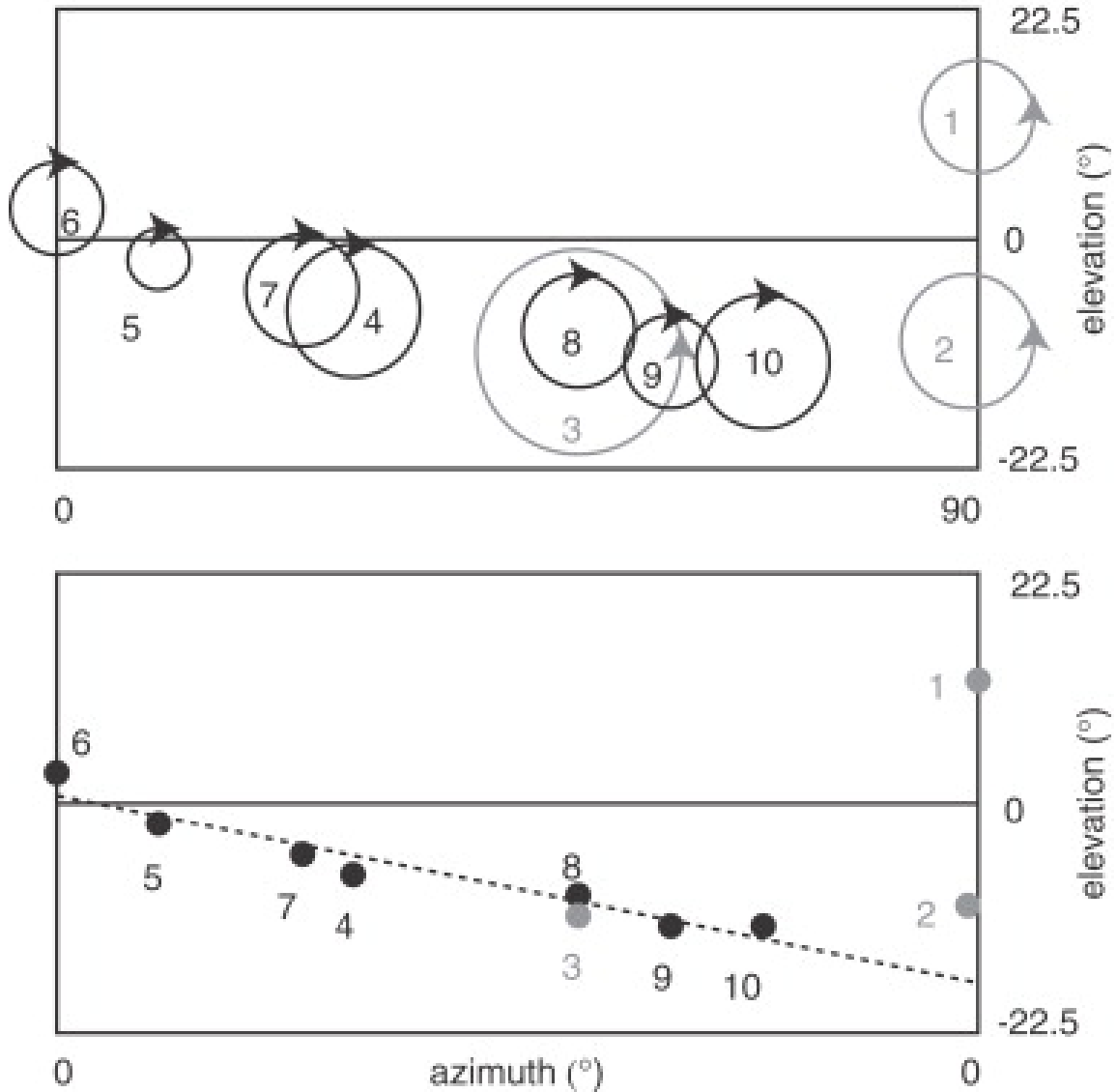


Figure 3.7: Preferred rotational axes of matched filters for VS cells of the right eye. The upper panel shows the preferred axes and sense of rotation of each VS cell (VS1-VS10) averaged over several individual flies. The diameter of circular arrows is directly proportional to standard deviation of axis of rotation across individual flies. Lower panel shows mean axis of rotation with VS4-VS10 shown to approximately lie on a line with 12° slope. Adapted from [4].

to pure roll motion while other VS cells on the line respond to some combination of yaw, pitch, and roll motion. VS1 and VS2 have separate axes locations and respond mainly to pitch motion with yaw to a lesser extent.

Not only is the direction to which VS-cells respond but the sensitivity to local motion information across the field is an indication of validity of the matched filter hypothesis. For instance, the VS-cells are more sensitive to local motion information in the dorsal region than the ventral region. This is an optimal strategy to produce a response that is based on pure rotational motion. This supports the argument that VS-cells mainly respond to rotational components of self-motion.

4. IMPLEMENTATION: AN AGENT MODEL

4.1 Overview

In this Section, I will give a detailed model and architecture of an agent that is capable of autonomously grounding the optical flow information by performing behaviorally meaningful actions. The agent's model is inspired from the fly's visuo-motor system that was discussed in Section 3. The agent is capable of sensing motion information from the surroundings and process it in a way that makes it useful for it to understand self-motion and other motions in the environment. It understands the meaning of motion information by learning a sensory state to motion mapping. With a computational model of the optical flow detectors in the fly, I will show that action (or coordinated motor output) is the only way that the fly can learn the meaning of its internal spikes and generate meaningful, relevant behavior, all just based on its internal spike patterns.

The next section describes the general model of the agent and the environment. In the subsequent sections, a detailed description of the agent architecture is provided.

4.2 Model of the Agent and the Environment

Consider a simple sensorimotor agent that is capable of sensing motion information in the form of optical flow and can perform a certain set of actions. The agent receives motion information from a moving scene which its local motion sensors transform into a global optical flow which is then picked up by motion detectors (modeling the fly's visual processing neurons) to produce spike pattern in the sensory array. The sensory array indicates the sensory state s which is used by agent to infer the semantics of the motion information and how it is related to the external environmental changes. The agent does not have direct access to external input I

or understands the functional properties of filters f . This situation is similar to Fig. 4.1, where external world properties need to be inferred based on internal spikes only. This means that the internal observer has to learn the meaning of s that corresponds to external stimuli I without having access to outside environment.

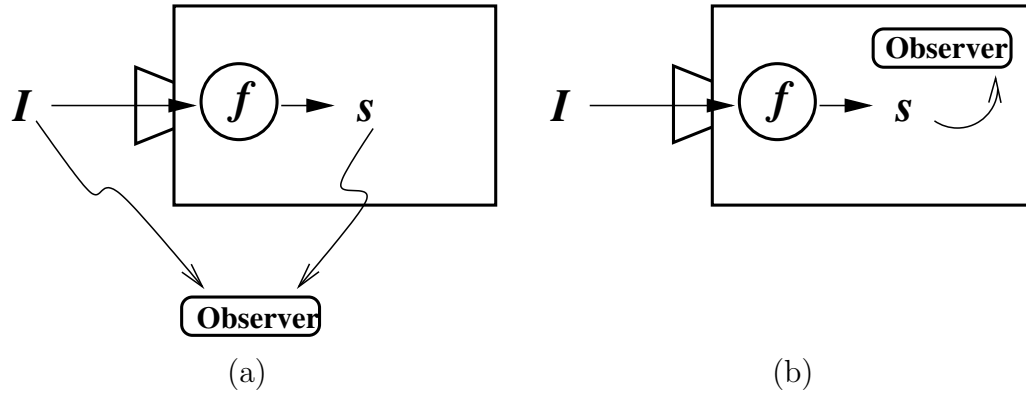


Figure 4.1: Model of the agent. The figure shows the problem of decoding internal sensory state from different point of views. (a) External Observer has access to both input I and state s , while (b) Internal Observer can only see internal states s . Here f are sensory filters that generate sensory state commensurate to external input. Adapted from [1].

Now assume that the agent receives an input in the form of optical flow generated by relative motion between the agent and the scene. The agent's sensors (modeled based on fly's visual system models) process this information and trigger a specific sensory state corresponding to the perceived optical flow. Based on this encoded state, how can the agent understand what the encoding means? A critical insight in previous work [1] [19] was that a specific pattern of action that maintain the internal state invariant is bound to have embody the same stimulus property encoded by the internal state. For example, if the perceived optical flow is rotate clockwise, rotating counterclockwise will keep the "rotate clockwise" state invariant over time

(the *invariance criterion*). This way, the agent can understand what these encoded internal states mean, in terms of its own actions. Below, I propose a sensorimotor learning technique based on reinforcement learning to learn this kind of sensor-action mapping with state invariance as the reward signal.

4.3 Computational Model of the Fly Visual System

In this section, I will describe the computational model of the agent’s optical flow detection system and the sensor-action mapping algorithm. I start with input processing, response generation and finally the learning rule that enables agent to learn the sensory state to motor response mapping.

4.3.1 Input Processing

The agent observes an input image (as seen in the figure on pg. 42) that is moving at a constant speed in a particular manner. For training, the image is moved to generate pitch, roll, yaw, and zooming (radiating) optical flows, in different instances with respect to the agent. These optical flows can be generated by rotating or translating the image along certain axes and directions with respect to the agent. The agent’s visual motion detectors based on the same principles as those of the fly’s EMDs detect this motion and produce a signal corresponding to the direction and velocity of the local motion. These detectors are spread across the entire visual field.

The following equations govern the optical flow calculation:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{4.1}$$

where intensity $I(x, y, t)$ at point (x, y, t) is moved by $\Delta x, \Delta y$ over time Δt . It is assumed that the intensity does not change over small magnitude of motion. Using

Taylor series expansion, we get,

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0 \quad (4.2)$$

which results in,

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0 \quad (4.3)$$

where $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}$ are derivatives of image at (x, y, t) in corresponding directions. This equation is the standard equation for optical flow calculation. This equation has two unknowns, V_x and V_y , velocity of point (x, y) . This is the situation where we have two unknowns but only one equation. This problem can only be solved by making additional assumptions. Two methods - Lucas-Kanade method [25] and Horn-Schunck method [26], both employ additional constraints that help solve this equation.

Outline of both the methods is given below:

1. **Lucas-Kanade Method:** It is a widely used method of optical flow estimation proposed by Bruce Lucas and Takeo Kanade in 1981. This method solves the ambiguity in optical flow equation by combining information from nearby pixels. This works by assuming that the flow is constant in local neighbourhood of pixel under consideration. Using this assumption, the method solves the optical flow equation for a group of adjacent pixels following the least squares criterion. This method has a low sensitivity to noise.
2. **Horn-Schunck Method:** This is a commonly used optical flow estimation method in computer vision. It was proposed by Berthold Horn and Brian Schunck in the year 1981. It is a global method as it applies global constraint of optical flow smoothness to solve the optical flow equation ambiguity. It

minimizes distortion in optical flow field by preferring smoothness over the entire field. This method is highly sensitive to noise.

These methods have been used in this thesis to generate optical flow fields because they are based on identical principles as fly’s Elementary Motion Detectors (EMDs) and are also widely used in field of computer vision. These differential methods give response that is linearly dependent on velocity and direction of motion. Also, these methods of calculating optical flow fields from moving images are compliant with the necessary and sufficient conditions that are required for analysis of directional motion, as discussed in Section 3.4.

I used MATLAB’s standard implementation of these methods for generating optical flow from images. The generated optical flow is normalized over the entire field of view using the l_2 -norm.

4.3.2 Sensory Primitives

The sensory state is generated by the modeled HS and VS cells using the optical flow generated by local motion sensors. The agent has two eyes that receive input from different regions in the visual field. Each eye, left and right, has its own set of mHS (Modeled HS) and mVS (Modeled VS) neurons. However, the overall sensory state is the output of all the neurons of both eyes, as they all act as inputs for the motor system. Consider that the agent has 3 mHS cells and 5 mVS neurons corresponding to each eye. These cells respond to inputs from their corresponding regions of the visual field (Tables 4.1 and 4.2). The visual field is 90° wide in elevation and 180° wide in azimuth. The sensitivities of HS and VS cells to motion depend upon the the spanning direction and density of the dendritic ramifications across the field (Fig. 3.6). The weighted response to optical flow by mHS and mVS neurons is given as a two dimensional Gaussian function as follows (after [13]). Following [14],

mHS	Elevation Range	Azimuth Range	(ϕ_m, θ_m)	σ_θ	$\sigma_{\phi+}$	$\sigma_{\phi-}$
Right mHSN	$[9^\circ, 45^\circ]$	$[-30^\circ, 90^\circ]$	$(11.25^\circ, 27^\circ)$	15°	75°	33.75°
Left mHSN	$[9^\circ, 45^\circ]$	$[-90^\circ, 30^\circ]$	$(-11.25^\circ, 27^\circ)$	15°	33.75°	75°
Right mHSE	$[-18^\circ, 18^\circ]$	$[-30^\circ, 90^\circ]$	$(11.25^\circ, 0^\circ)$	15°	75°	33.75°
Left mHSE	$[-18^\circ, 18^\circ]$	$[-90^\circ, 30^\circ]$	$(-11.25^\circ, 0^\circ)$	15°	33.75°	75°
Right mHSS	$[-45^\circ, -9^\circ]$	$[-30^\circ, 90^\circ]$	$(11.25^\circ, -27^\circ)$	15°	75°	33.75°
Left mHSS	$[-45^\circ, -9^\circ]$	$[-90^\circ, 30^\circ]$	$(-11.25^\circ, -27^\circ)$	15°	33.75°	75°

Table 4.1: Parameter values of modeled HS cells (mHS).

we can model the mHS neurons as:

$$\begin{aligned}
w(\phi, \theta) &= \exp\left(-\left(\frac{\theta - \theta_m}{\sigma_\theta}\right)^2 - \left(\frac{\phi - \phi_m}{\sigma_{\phi+}}\right)^2\right), \text{ if } \phi \geq \phi_m \\
&= \exp\left(-\left(\frac{\theta - \theta_m}{\sigma_\theta}\right)^2 - \left(\frac{\phi - \phi_m}{\sigma_{\phi-}}\right)^2\right), \text{ otherwise,}
\end{aligned} \tag{4.4}$$

where θ is the elevation angle, ϕ is the azimuth angle, (ϕ_m, θ_m) is the center of receptive field for that neuron (Fig. 4.2). The angular width for elevation is σ_θ while for azimuth it is $\sigma_{\phi+}$ and $\sigma_{\phi-}$. The parameter values for all 6 mHS neurons are shown in Table 4.1.

For mVS neurons (Fig. 4.2),

$$\begin{aligned}
w(\phi, \theta) &= \exp\left(-\left(\frac{\theta - \theta_m}{\sigma_{\theta+}}\right)^2 - \left(\frac{\phi - \phi_m}{\sigma_\phi}\right)^2\right), \text{ if } \theta \geq \theta_m \\
&= \exp\left(-\left(\frac{\theta - \theta_m}{\sigma_{\theta-}}\right)^2 - \left(\frac{\phi - \phi_m}{\sigma_\phi}\right)^2\right), \text{ otherwise,}
\end{aligned} \tag{4.5}$$

where θ is the elevation angle, ϕ is the azimuth angle, (ϕ_m, θ_m) is the center of receptive field for that neuron. The angular width for azimuth is σ_ϕ while for elevation it is $\sigma_{\theta+}$ and $\sigma_{\theta-}$. The parameter values for all 10 mVS neurons is shown in Table 4.2.

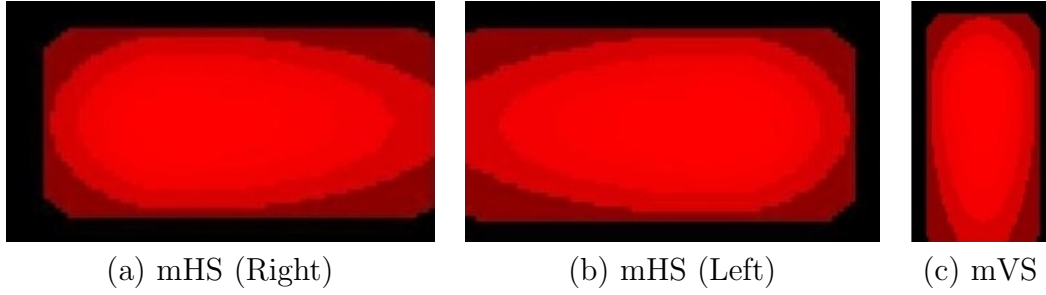


Figure 4.2: Weight distribution of mVS and mHS neurons. The 2-D Gaussian plots show weight distribution of mVS and mHS neurons across the field based on dendrite distribution and density pattern (Fig. 3.6).

Finally, we normalize the weighted response by the sum:

$$w(\phi, \theta) = \frac{w(\phi, \theta)}{\sum_{\phi, \theta} w(\phi, \theta)}. \quad (4.6)$$

Each model neuron receives optical flow input from a region that they respond to as shown in Tables 4.1 and 4.2. The mHS neurons give positive response to horizontal motion in direction from center to side of the field, which is its preferred direction. They give negative response for motion in the opposite direction. The mVS neurons' preferred direction is up to down. Each of these cells integrate the local motion information to generate a response commensurate with the global picture. The response of all neurons form a column vector s' with each element s'_i corresponding to an individual neuron's response. s'_i is the weighted sum of net motion in the neuron's preferred direction as given below:

1. Method I.

$$s'_i = v_{i+} - v_{i-} \quad (4.7)$$

mVS	Azimuth Range	(ϕ_m, θ_m)
Right mVS1	$[-30^\circ, 0.4^\circ]$	$(-14.8^\circ, 15^\circ)$
Left mVS1	$[-0.4^\circ, 30^\circ]$	$(14.8^\circ, 15^\circ)$
Right mVS2	$[-7.6^\circ, 22.8^\circ]$	$(7.6^\circ, 15^\circ)$
Left mVS2	$[-22.8^\circ, 7.6^\circ]$	$(-7.6^\circ, 15^\circ)$
Right mVS3	$[14.8^\circ, 45.2^\circ]$	$(30^\circ, 15^\circ)$
Left mVS3	$[-45.2^\circ, -14.8^\circ]$	$(-30^\circ, 15^\circ)$
Right mVS4	$[37.2^\circ, 67.6^\circ]$	$(52.4^\circ, 15^\circ)$
Left mVS4	$[-67.6^\circ, -37.2^\circ]$	$(-52.4^\circ, 15^\circ)$
Right mVS5	$[59.6^\circ, 90^\circ]$	$(74.8^\circ, 15^\circ)$
Left mVS5	$[-90^\circ, -59.6^\circ]$	$(-74.8^\circ, 15^\circ)$

Table 4.2: Parameter values of modeled VS cells (mVS). Some parameters that are common among all mVS are: Elevation Range= $[-45^\circ, 45^\circ]$, $\sigma_\phi = 24^\circ$, $\sigma_{\theta+} = 24^\circ$ and $\sigma_{\theta-} = 37.5^\circ$.

where

$$v_{ix} = \sum_{\phi, \theta} w_i(\phi, \theta) m_x(\phi, \theta). \quad (4.8)$$

Here $i = 16$ is the total number of neurons, $x = (+/-)$, $w_i(\phi, \theta)$ is the weight distribution of i^{th} neuron, $m_x(\phi, \theta)$ is the normalized optical flow in neuron's preferred direction ($x = +$) or null direction ($x = -$).

- Method II.** In this method, the property of VS cells to respond to motion matched to a particular type of rotational motion is used (see Section 3.8). VS cells respond to rotation along axes as shown in Fig. 3.7. Using this property, mVS cells are modeled to respond to motion along axes as shown in Table 4.3. The response of mVS cells in their respective region of response (Table 4.1 and 4.2) is along rotational fields about these axes with up to down direction as the preferred direction. The mHS cell model is the same as in Method I.

$$s'_i = v_{i+} - v_{i-} \quad (4.9)$$

mVS	Center of Axes (ϕ_c, θ_c)
Right mVS1	$(90^\circ, 15^\circ)$
Left mVS1	$(-90^\circ, 15^\circ)$
Right mVS2	$(90^\circ, -15^\circ)$
Left mVS2	$(-90^\circ, -15^\circ)$
Right mVS3	$(0^\circ, 0^\circ)$
Left mVS3	$(0^\circ, 0^\circ)$
Right mVS4	$(22.5^\circ, -2^\circ)$
Left mVS4	$(-22.5^\circ, -2^\circ)$
Right mVS5	$(45^\circ, -4^\circ)$
Left mVS5	$(-45^\circ, -4^\circ)$

Table 4.3: Matched filter parameters of modeled VS cells (mVS). The table gives the center of axes of rotation along which mVS cells are matched to respond.

where

$$\begin{aligned}
v_{ix} &= \sum_{\phi, \theta} w_i(\phi, \theta) m_x(\phi, \theta), \text{ for mHS neurons} \\
&= \sum_{\phi, \theta} w_i(\phi, \theta) (op(\phi, \theta) \cdot mat_{i,x}(\phi_c, \theta_c)), \text{ for mVS neurons}
\end{aligned} \tag{4.10}$$

Here $i = 16$ is the total number of neurons, $x = (+/-)$, $w_i(\phi, \theta)$ is the weight distribution of i^{th} neuron, $m_x(\phi, \theta)$ is the normalized optical flow in neuron's preferred direction ($x = +$) or null direction ($x = -$). $mat_i(\phi_c, \theta_c)$ is the matched filter mask (rotational along center (ϕ_c, θ_c) as shown in Table 4.3) of i^{th} neuron, mVS neuron, in its preferred direction ($x = +$) or null direction ($x = -$). $op(\phi, \theta)$ is the global optical flow.

The response vector s' is normalized using its l_2 -norm:

$$s' = \frac{s'}{|s'|}. \tag{4.11}$$

The agent then compares this generated response to the reference responses s'_{ref} which are the ideal standard optical flows (Fig. 2.2)- yaw, pitch, roll, zoom (radiate):

$$s = \arg \max_{\psi_i, i=1\dots n} (s' \cdot s'_{\text{ref}(i)}) \quad (4.12)$$

where s is the sensory state that corresponds to $i = 8$ types of optical flows. This sensory state is what the agent is trying to figure out the meaning of.

4.3.3 Learning Algorithm

The reinforcement learning algorithm to learn the sensory state to action mapping is based on previous work [19]. Consider the sensory state s_t at time t . The agent performs an action a_t , which leads to a change in the sensory state to s_{t+1} at time $t+1$. This transition of state really depends on the net optical flow generated between agent and scene. As discussed, the agent wants to minimize the variation in state (invariance criterion) while performing actions. This means that the reward that the agent gets should be inversely proportional to variation in current state s_t . The immediate reward ρ_{t+1} is calculated as,

$$\rho_{t+1} = \frac{1}{\sqrt{\sum_i (r_{t+1,i} - r_{t,i})^2}} \quad (4.13)$$

where $r_{t,i}$ is the i^{th} element of the sensory response vector r_t at time t . Here the reward is calculated as the inverse of the Euclidean distance between the current state response vector r_t and the next state response vector r_{t+1} , as opposed to using simple binary values of variant or non-variant. Since the denominator in the equation can approach zero, leading to very high ρ_{t+1} , an upper bound of $\rho_{\text{max}} = 100$ is enforced.

Now, the agent has to learn a state-action mapping such that the reward ρ_t is maximized at time t . The agent has to learn to take action a_t such that the variation

in current state (s_t) to next state (s_{t+1}) is minimized. Since, the state transition is probabilistic (to avoid greedy exploitation) so the problem is to determine $P(a_t|s_t)$, which is conditional probability of taking action a_t at state s_t resulting in s_{t+1} that is highly likely to be same as s_t . Let's call this the reward probability function $R(s_t, a_t)$ (Fig. 4.3). The learning algorithm below is followed by the agent:

1. Initialize the $R(s, a)$ table randomly.
2. Given the current state is $s_t \in S$, randomly perform action $a_t \in A$.
3. For initial 200 iterations
 - (a) Perform action a_t .
 - (b) Otherwise, if a_t is $\arg \max_{a \in A} R(s_t, a)$
 - i. Then perform action a_t
 - ii. else perform action a_t with probability $R(s_t, a_t)$.
4. Repeat 2 and 3 until exactly one action is performed.

During the initial 200 iterations the learning algorithm selects the actions uniformly in random fashion. This is done to ensure that no action gets preference due to randomly generated initial values of reward probability function table. The $R(s, a)$ entries (Fig. 4.3) are updated after performing selected actions every iteration. This is done as follows:

$$R_{t+1}(s_t, a_t) = R_t(s_t, a_t) + \alpha \rho_{t+1} \quad (4.14)$$

where $R_t(., .)$ is reward probability function at time t , α is the learning rate ($\alpha = 0.001$). Finally, the reward table is normalized,

$$R_{t+1}(s_t, a) = \frac{R_{t+1}(s_t, a)}{\sum_{a_i \in A} R_{t+1}(s_t, a_i)}. \quad (4.15)$$









$R(s, a)$	Direction of Motion (a)							
								
RYRL	1	0	0	0	0	0	0	0
RYLR	0	1	0	0	0	0	0	0
RPUD	0	0	1	0	0	0	0	0
RPDU	0	0	0	1	0	0	0	0
RRCL	0	0	0	0	1	0	0	0
RRAC	0	0	0	0	0	1	0	0
TLRI	0	0	0	0	0	0	1	0
TLRO	0	0	0	0	0	0	0	1

Figure 4.3: Reward table $R(s, a)$ with ideal reward values.

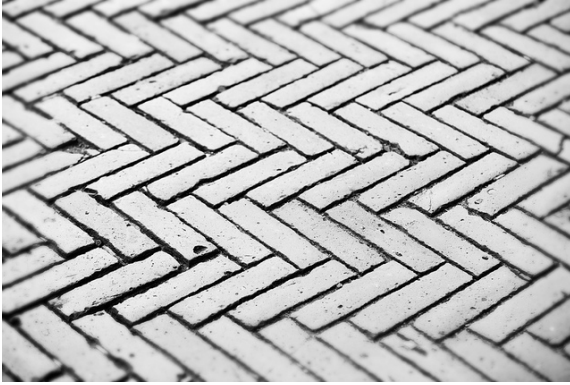
This is done for every action (a).

5. EXPERIMENTS AND RESULTS

5.1 Experimental Setup

In order to test the effectiveness of the learning algorithm as mentioned in Section 4, I conducted experiments to study the nature of state-action mapping obtained using the given models. The experiments were conducted with different sets of parameter values in the model over various images - synthetic and natural scenes. I experimented with the Lucas-Kanade method and the Horn-Schunck method for optical flow computation, where both gave similar results with slight variation based on the input image's characteristics. Different values for learning rate (α), modeled neuron's parameters ($\sigma_\theta, \sigma_\phi$) were tested. I also tried various techniques for generating reward (ρ) such as dot product, as well as the formula in Eq. 4.13.

Here I show results for 4 input images as shown in Fig. 5.1. The first one is a high contrast-low texture synthetic scene, the second a low contrast-high texture indoor scene, the third a low contrast-high texture natural scene, and the fourth a high contrast-high texture natural scene. These training images were converted to gray scale and scaled to a resolution of [100,100]. The agents were trained using 8 sensory primitives and 8 motor primitives (Fig. 4.3). The training was done for 500 iterations per sensory primitive, where the input image was moved in a particular fashion to generate that specific sensory primitive. The learning algorithm was executed to learn a state-action mapping for this state. This procedure was repeated for all 8 sensory primitives. Therefore, overall the training involved a minimum of 4000 iterations to get good results.



(a)



(b)



(c)



(d)

Figure 5.1: Input images. (a) Synthetic image with high contrast and low texture. (b) Indoor image with low contrast and high texture. (c) Natural image with low contrast and high texture. (d) Natural image with high contrast and high texture.

5.2 Results

To quantitatively measure the performance of the algorithm, I compared the Ideal R-table ($R_I(s, a)$) and the learned R-table ($R_L(s, a)$) over the training iterations.

This was done in the following manner:

$$E = \sum_{s,a} |R_L(s, a) - R_I(s, a)|, \quad (5.1)$$

where $|\cdot|$ represents absolute value. The error (E) represents the discrepancy between the ideal R-table and learned R-table. This error is calculated after every iteration in the learning algorithm and is shown to decrease to near zero over the iterations, for all four training images (Fig. 5.1). The error at the end of the training process approaches zero, meaning that the learning algorithm works as expected. The reward table $R(s, a)$ was initially randomly generated before training (Fig. 5.2(a)). The expected ideal R-table is a identity matrix of size 8 (Fig. 5.2(b)).

In following subsections, I show the results for different scenarios of inputs and parameters. They show how the output is affected by change in parameters associated with the agent model and the learning method.

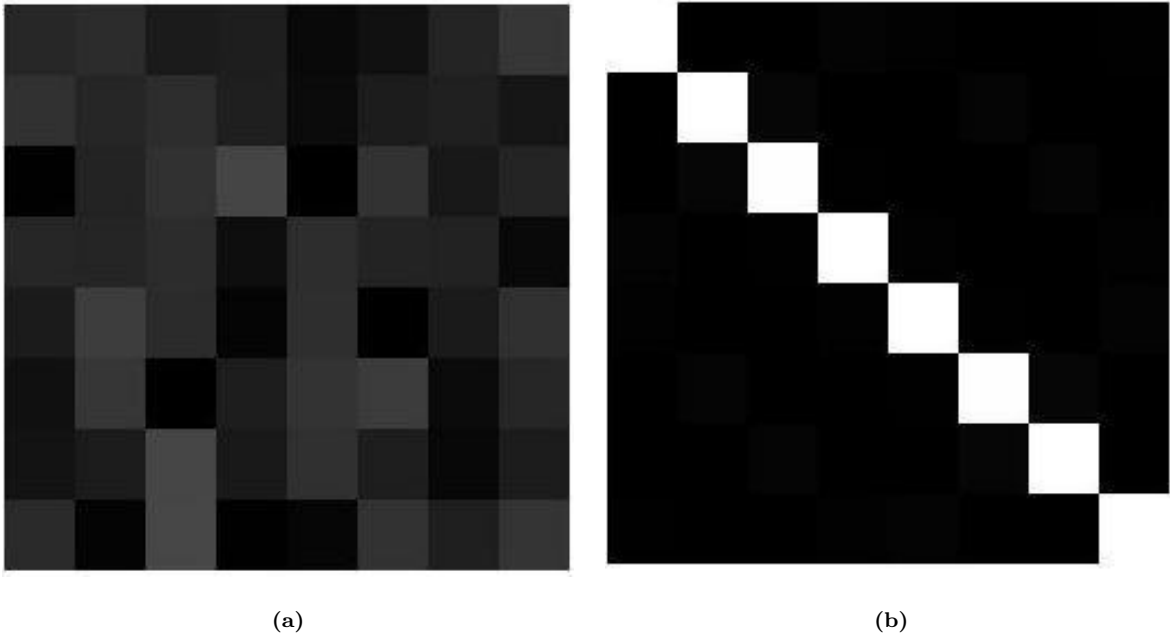


Figure 5.2: Initial and expected ideal reward table $R(s, a)$. (a) Initial randomly generated reward table. (b) Expected Ideal final reward table (Identity Matrix).

5.2.1 Scenario 1

Consider the agent model described in Section 4 and the basic experimental setting as highlighted in Section 5.1. This basic setting is used with Method I for calculating the sensory state, as presented in Subsection 4.3.2. The learning algorithm was trained using the reward calculated as the inverse of Euclidean distance of sensory response over time as shown in Equation 4.13 with a learning rate of $\alpha = 0.001$.

Fig. 5.3 shows the final reward tables obtained after training with these parameters. The progression of error over the training iterations is shown in Fig. 5.4. Finally, Fig. 5.5, 5.6, 5.7, and 5.8 show the choice of actions made by the agent during the training phase. A total of 4000 iterations were divided into 8 plots showing 500 iter-

ations each. Actions: 1=Yaw Left to Right, 2=Yaw Right to Left, 3=Pitch Down to Up, 4=Pitch Up to Down, 5=Roll Anti-Clockwise, 6=Roll Clockwise, 7=Translation Move Forward (Zoom In), 8=Translation Move Back (Zoom Out).

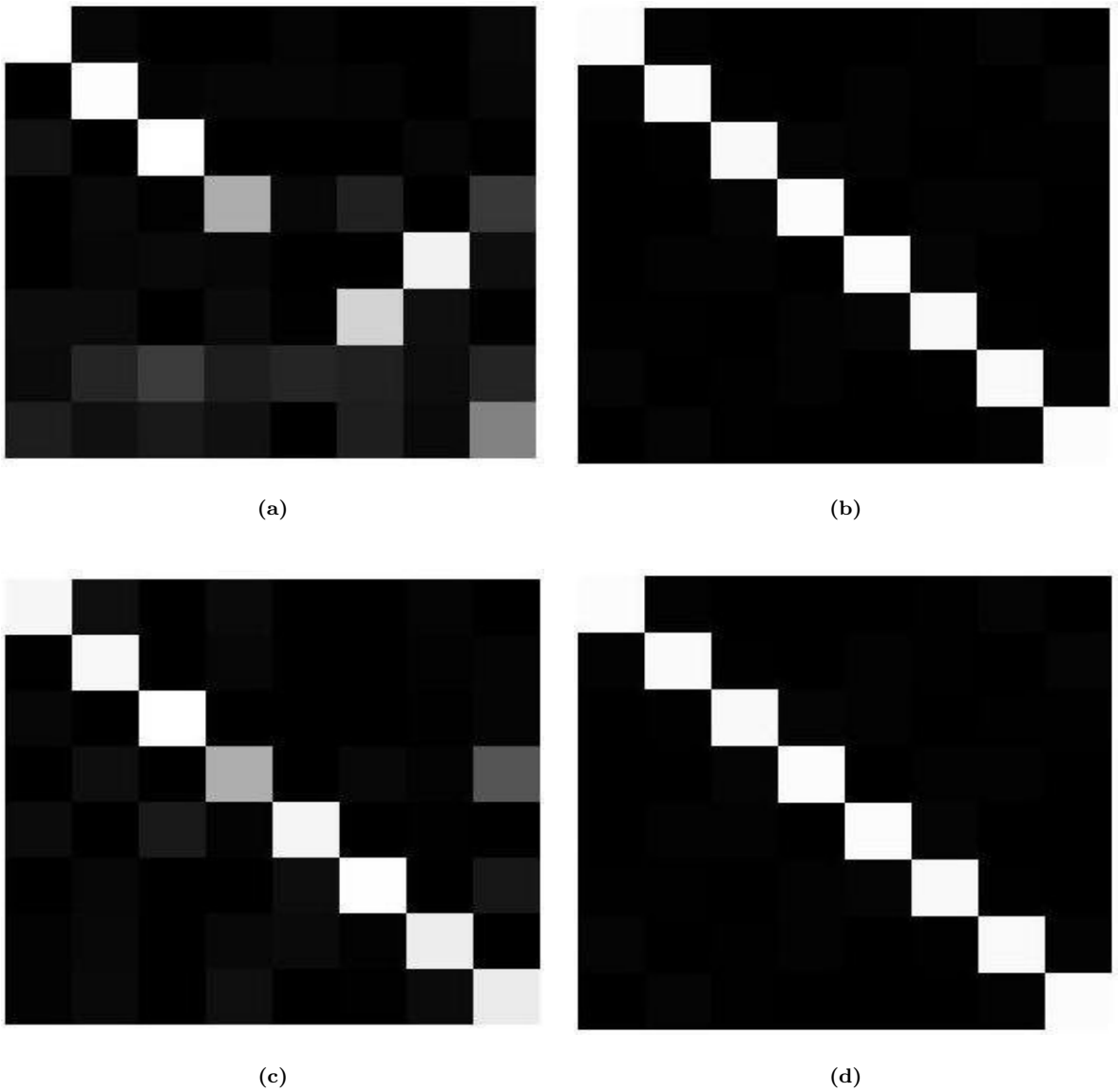


Figure 5.3: Final reward table $R(s, a)$ in Scenario 1. Final reward table obtained after training for Scenario 1 on the input images (in the same order as in Fig. 5.1).

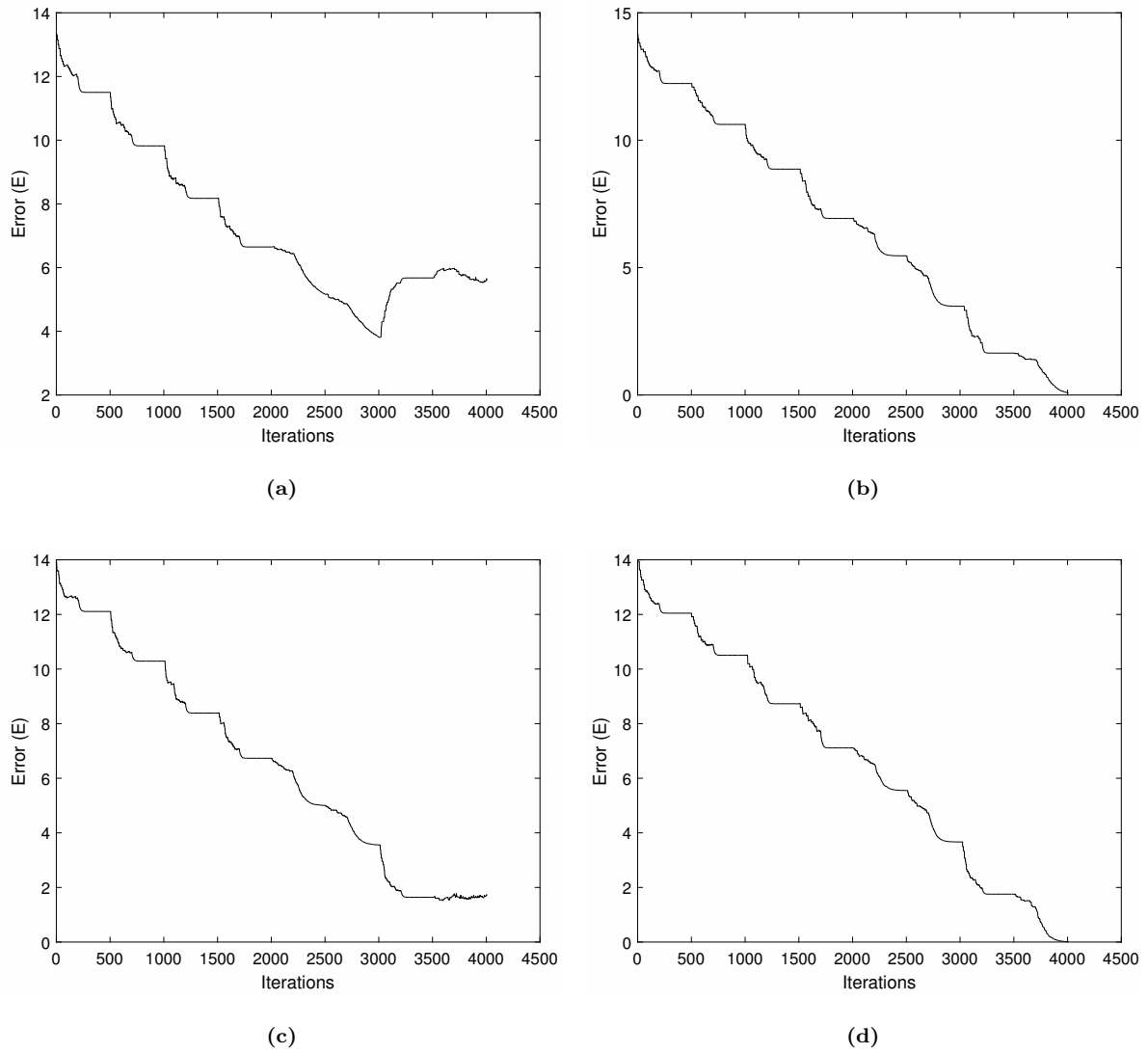


Figure 5.4: Error plot in Scenario 1. Shows progression of error over the training iterations for Scenario 1. The error is calculated based on Equation 5.1. The error plots are generated for the four input images (in the same order as in Fig. 5.1).

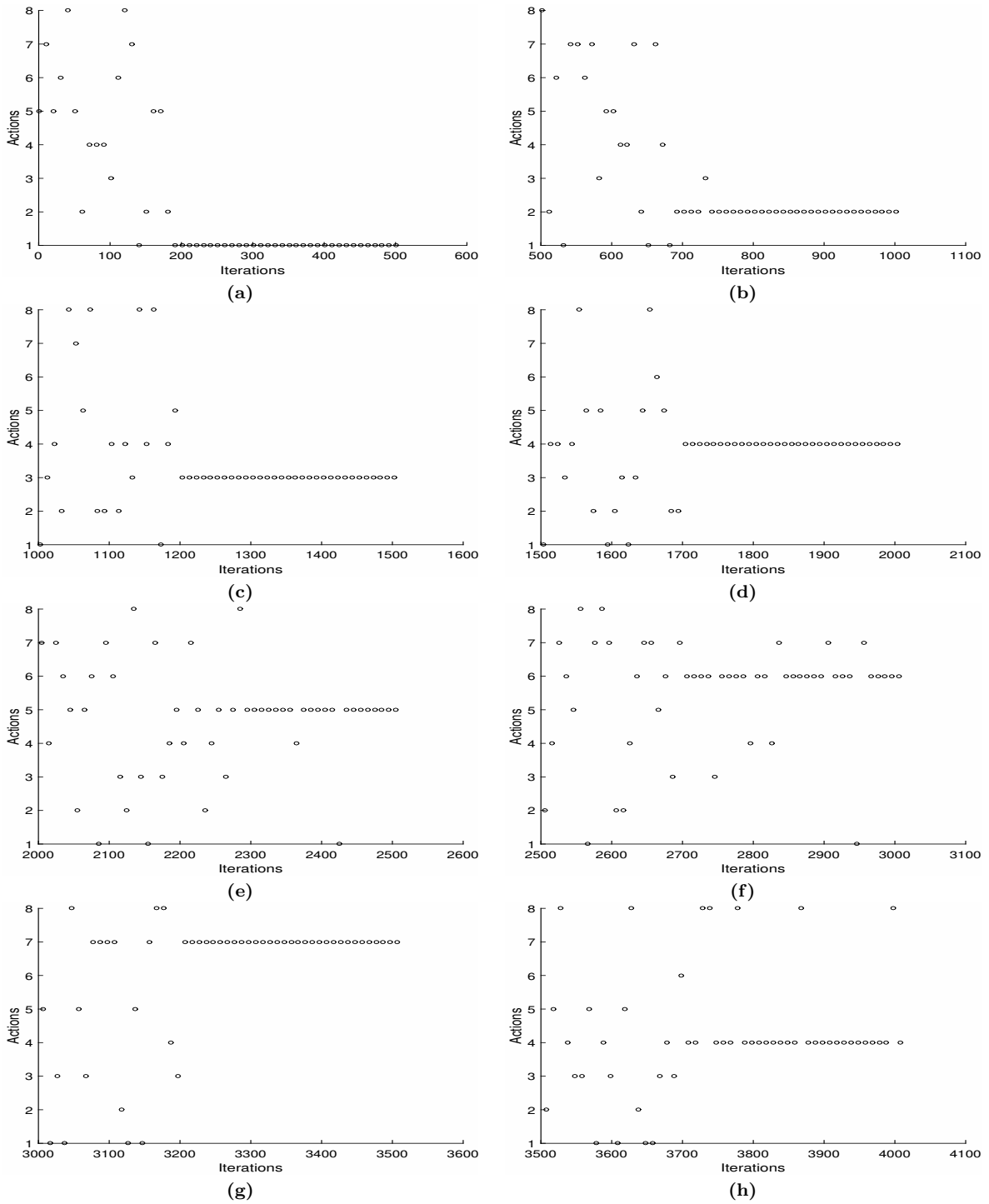


Figure 5.5: Progress plot for input image Fig. 5.1(a) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(a) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

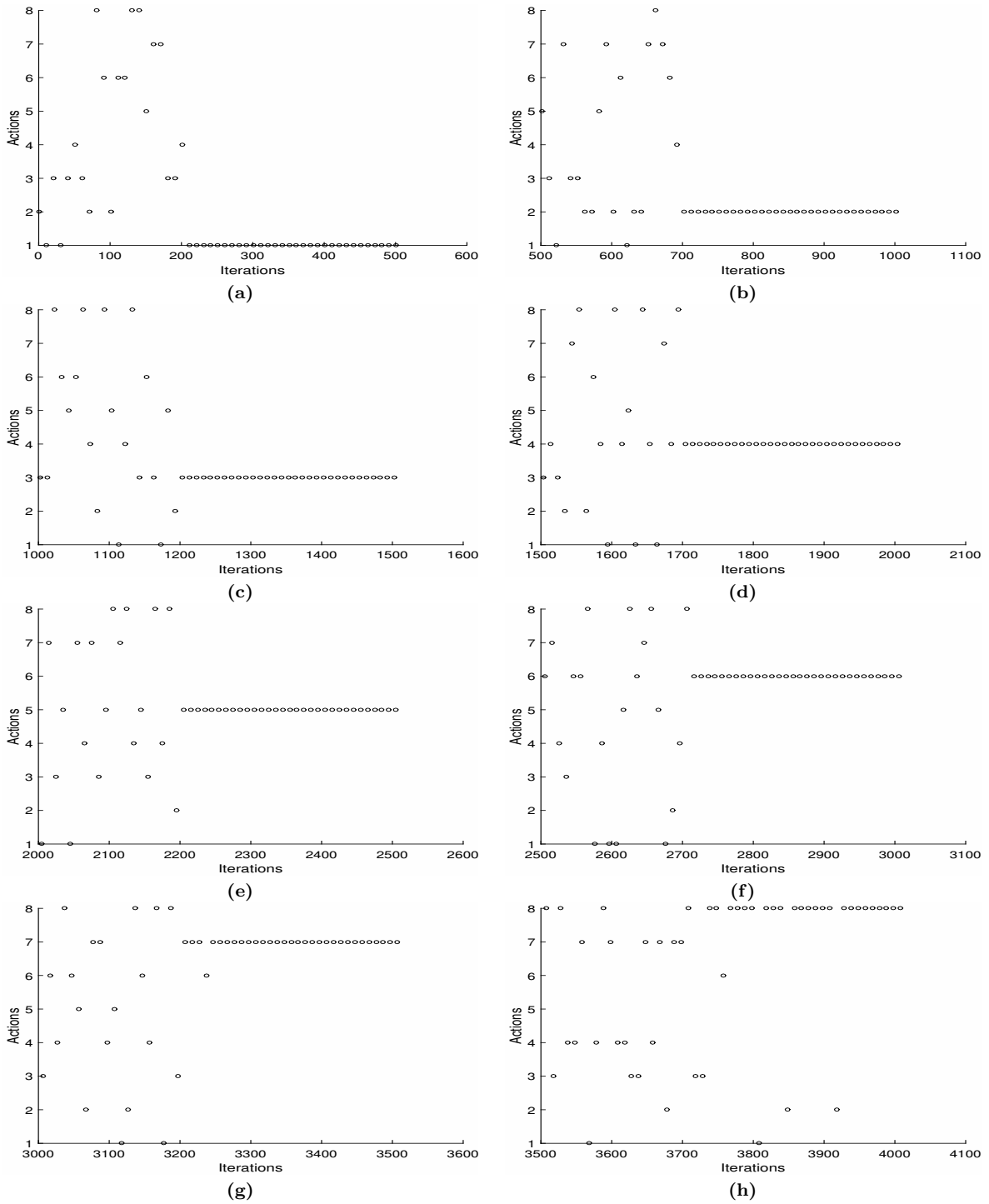


Figure 5.6: Progress plot for input image Fig. 5.1(b) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(b) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

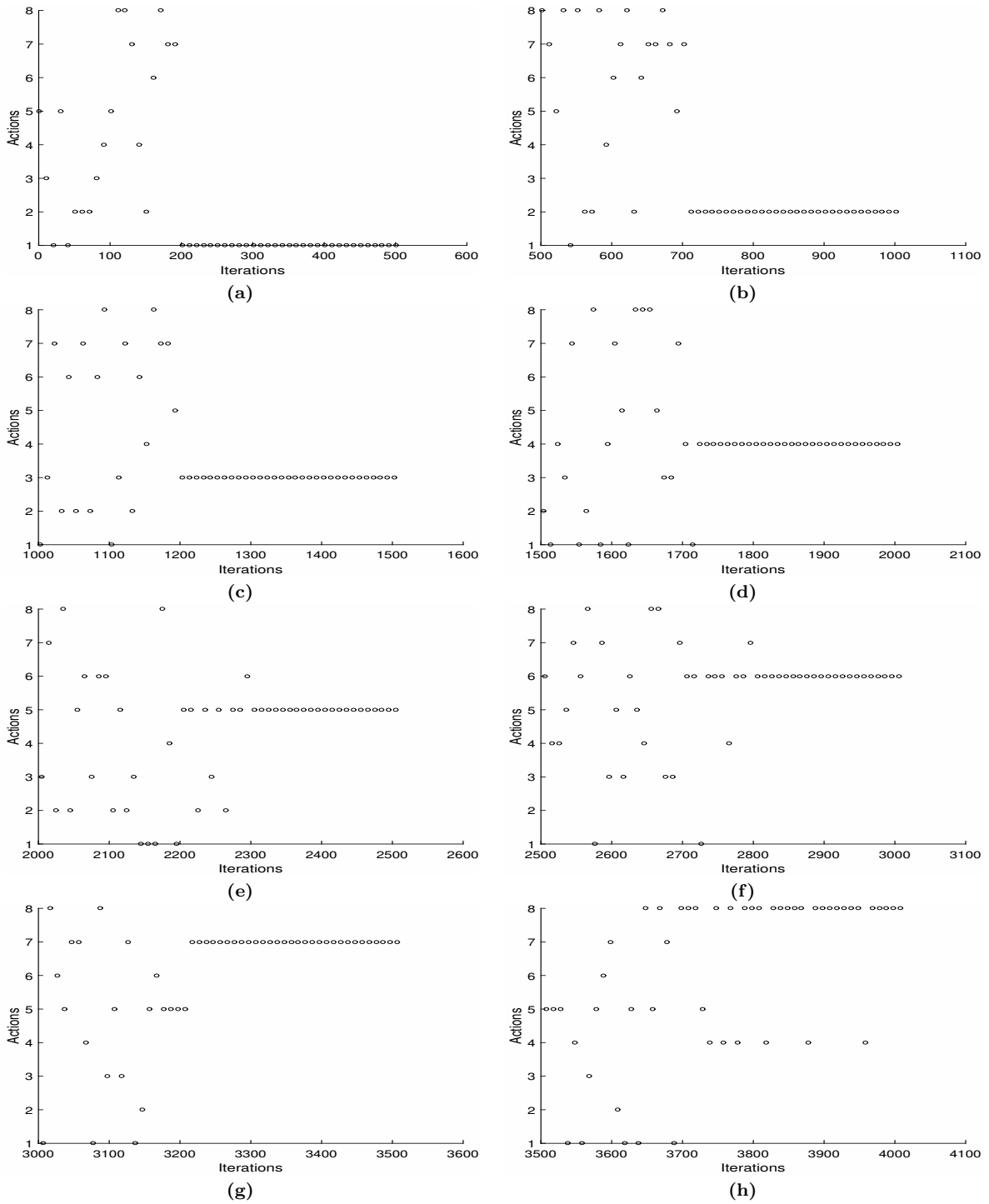


Figure 5.7: Progress plot for input image Fig. 5.1(c) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(c) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

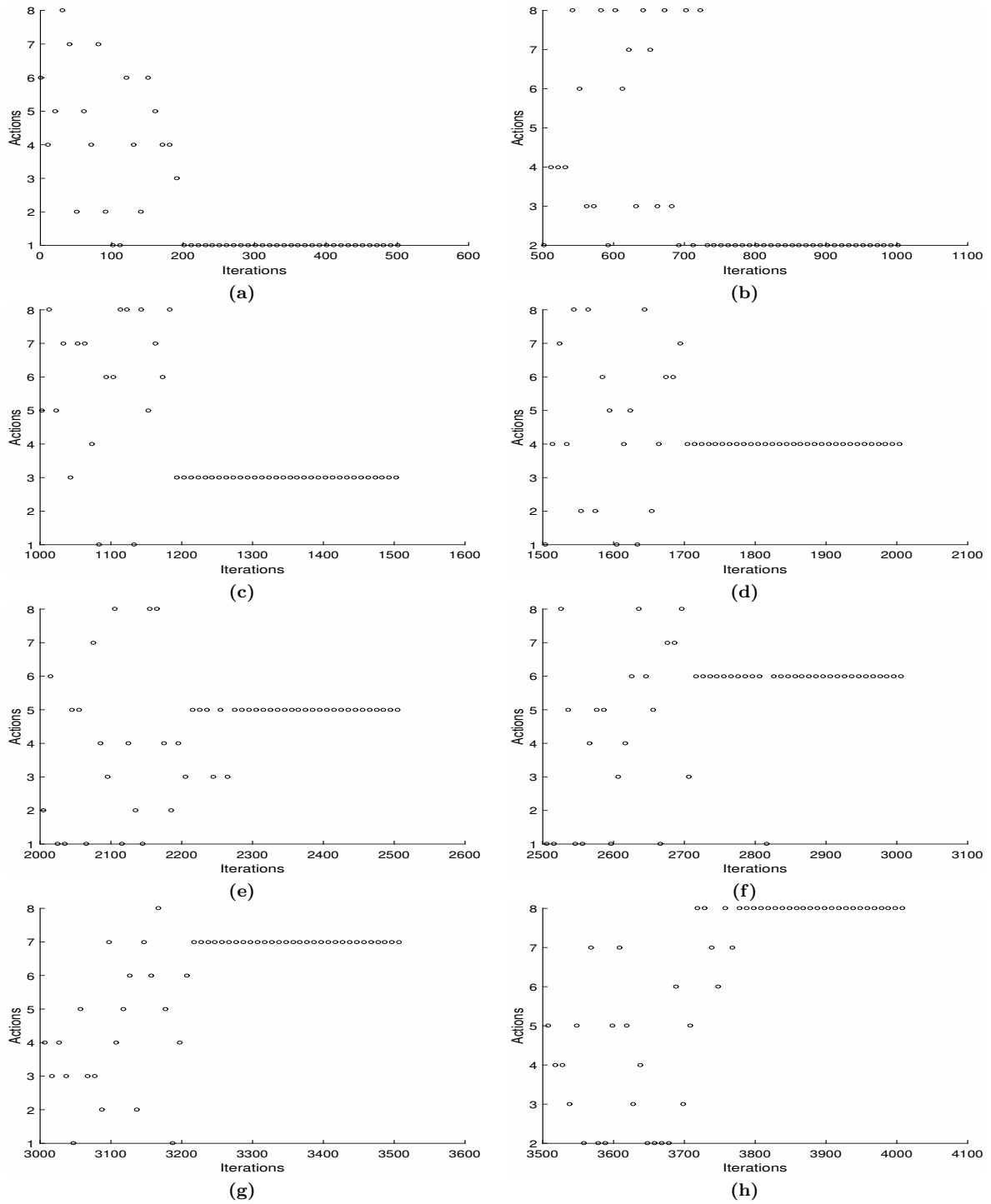
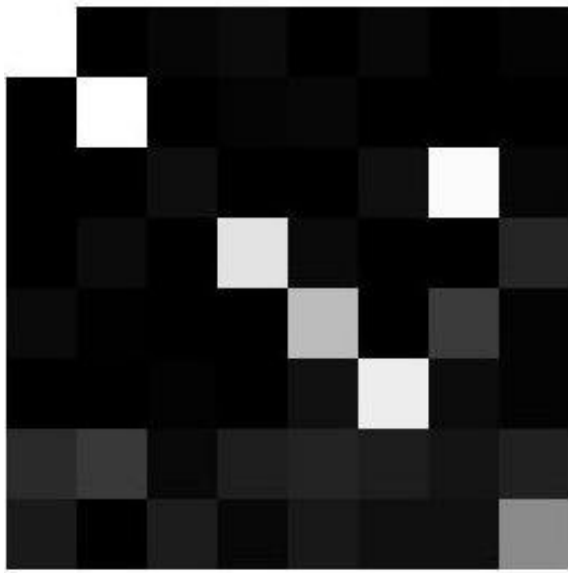


Figure 5.8: Progress plot for input image Fig. 5.1(d) in Scenario 1. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(d) for Scenario 1. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

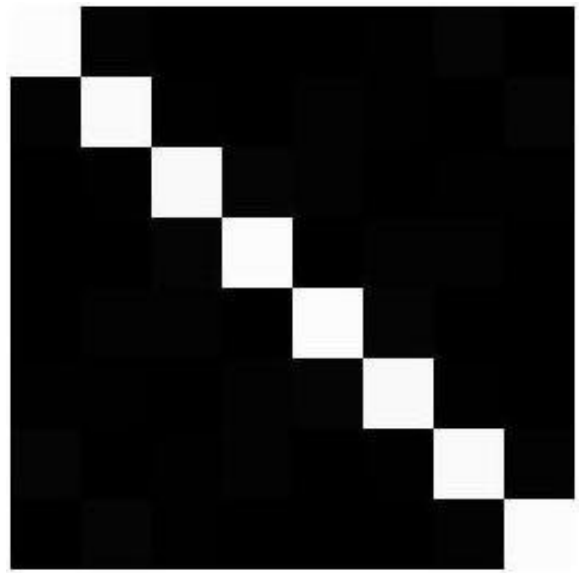
5.2.2 Scenario 2

Consider the agent model described in Section 4 and the basic experimental setting as highlighted in Section 5.1. This basic setting is used with Method II, using matched filter model of VS cells for calculating the sensory state, as presented in Subsection 4.3.2. The learning algorithm was trained using the reward calculated as the inverse of Euclidean distance of sensory response over time as shown in Equation 4.13 with a learning rate of $\alpha = 0.001$.

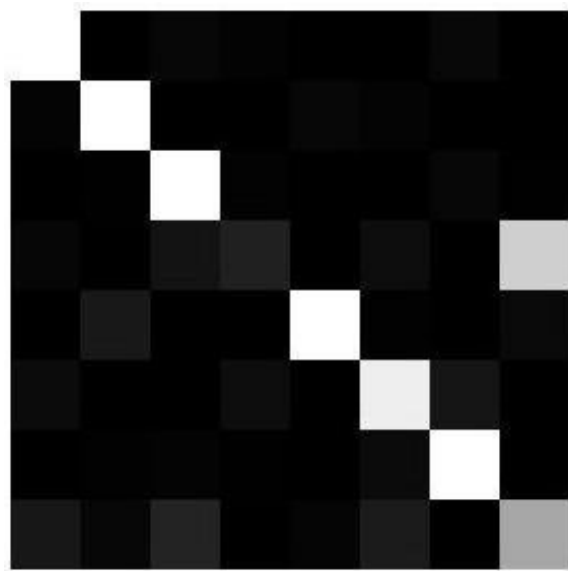
Fig. 5.9 shows the final reward tables obtained after training with these parameters. The progression of error over the training iterations is shown in Fig. 5.10. Finally, Fig. 5.11, 5.12, 5.13, and 5.14 shows the choice of actions made by the agent during the training phase. A total of 4000 iterations were divided into 8 plots showing 500 iterations each. Actions: 1=Yaw Left to Right, 2=Yaw Right to Left, 3=Pitch Down to Up, 4=Pitch Up to Down, 5=Roll Anti-Clockwise, 6=Roll Clockwise, 7=Translation Move Forward (Zoom In), 8=Translation Move Back (Zoom Out).



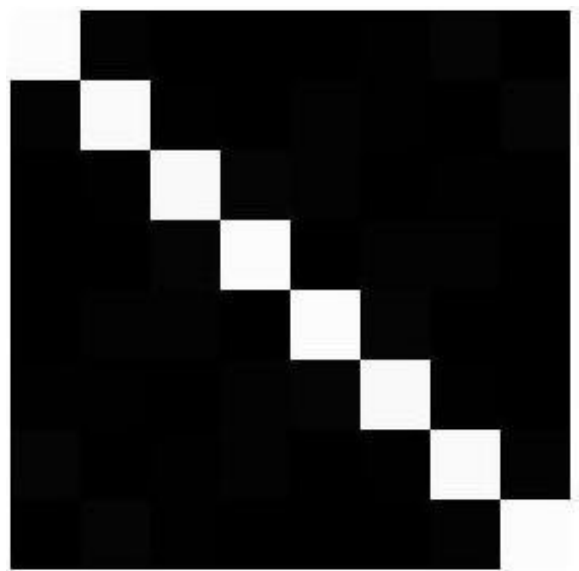
(a)



(b)

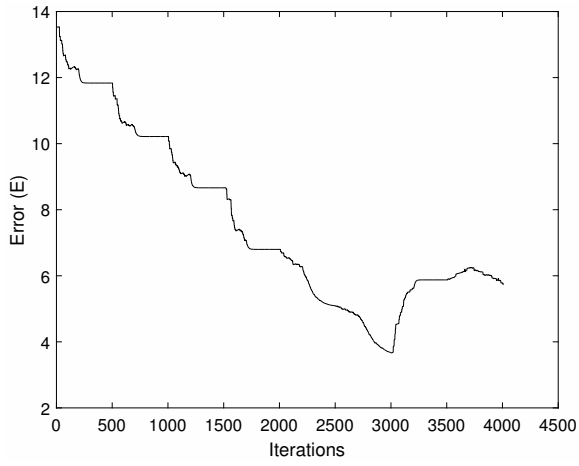


(c)

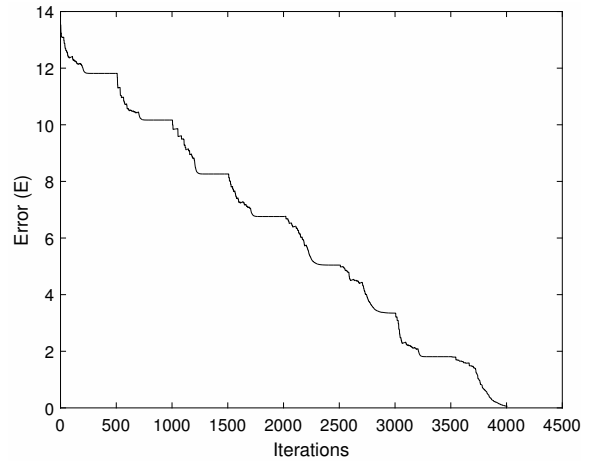


(d)

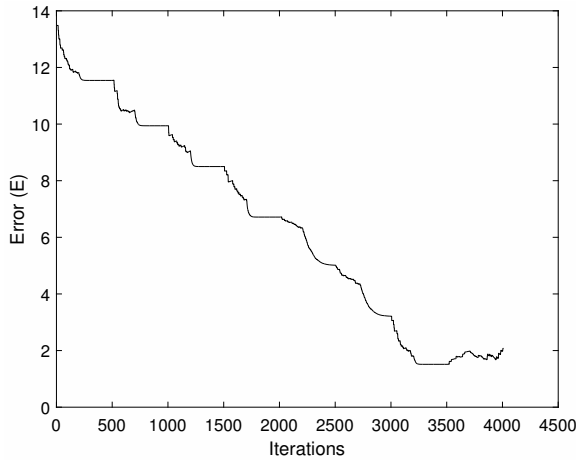
Figure 5.9: Final reward table $R(s, a)$ in Scenario 2. Final reward table obtained after training for Scenario 2 on the input images (in the same order as in Fig. 5.1).



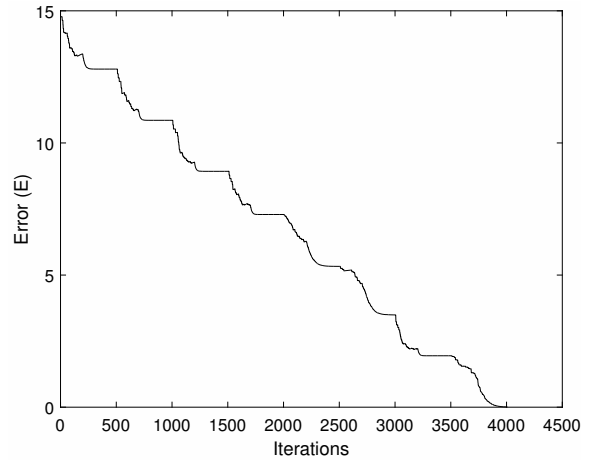
(a)



(b)



(c)



(d)

Figure 5.10: Error plot in Scenario 2. Shows progression of error over the training iterations for Scenario 2. The error is calculated based on Equation 5.1. The error plots are generated for the four input images (in the same order as in Fig. 5.1).

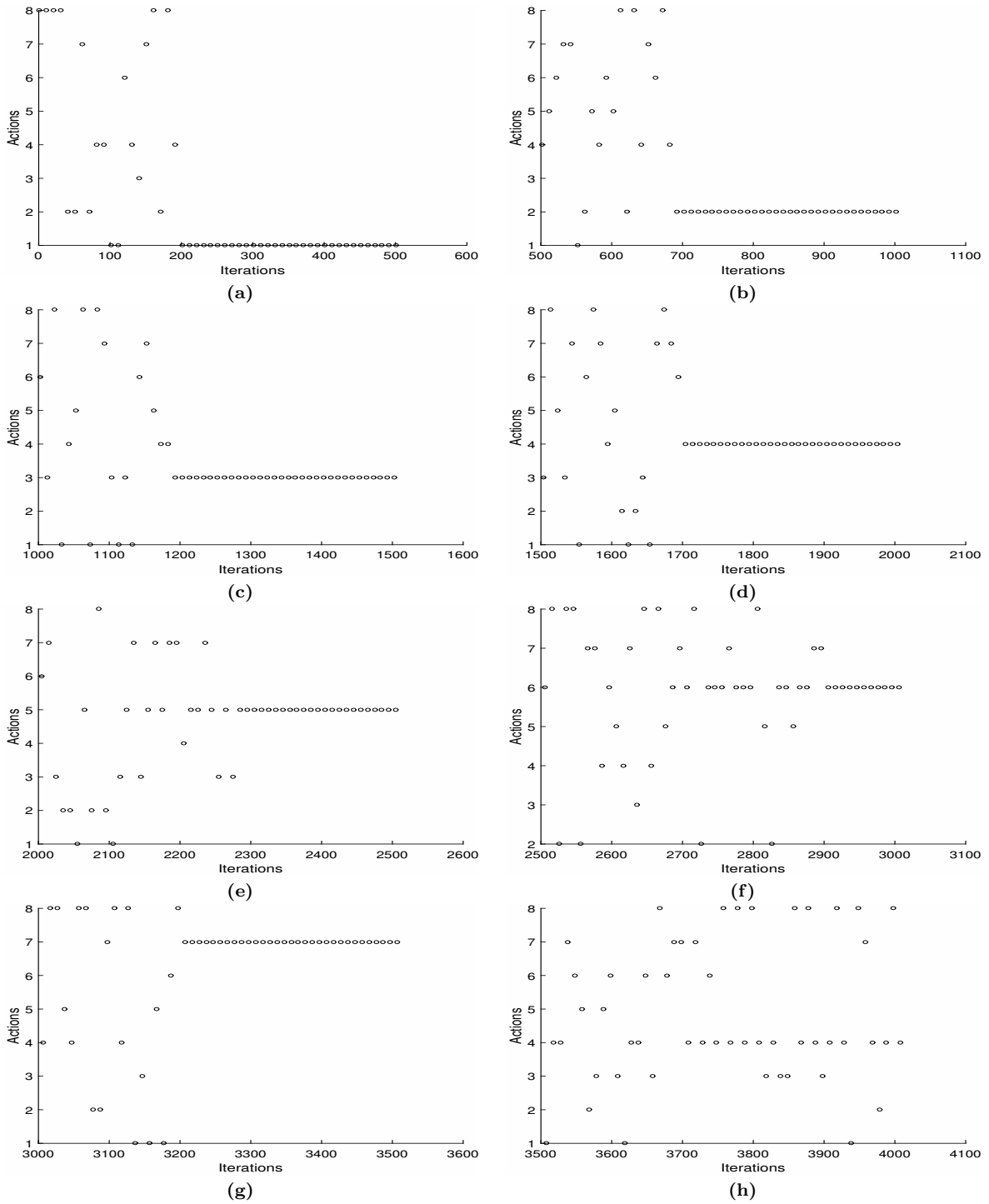


Figure 5.11: Progress plot for input image Fig. 5.1(a) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(a) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

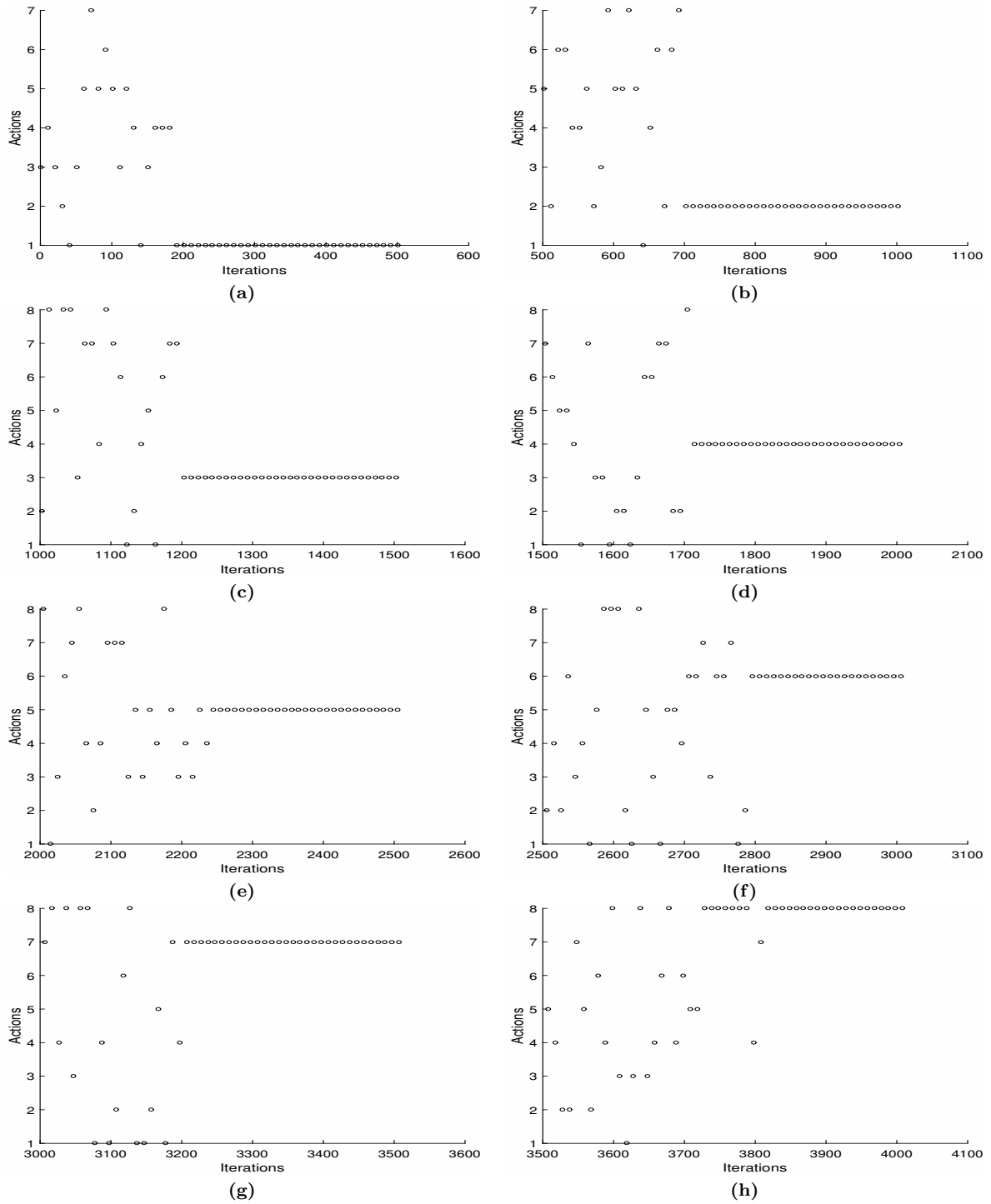


Figure 5.12: Progress plot for input image Fig. 5.1(b) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(b) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

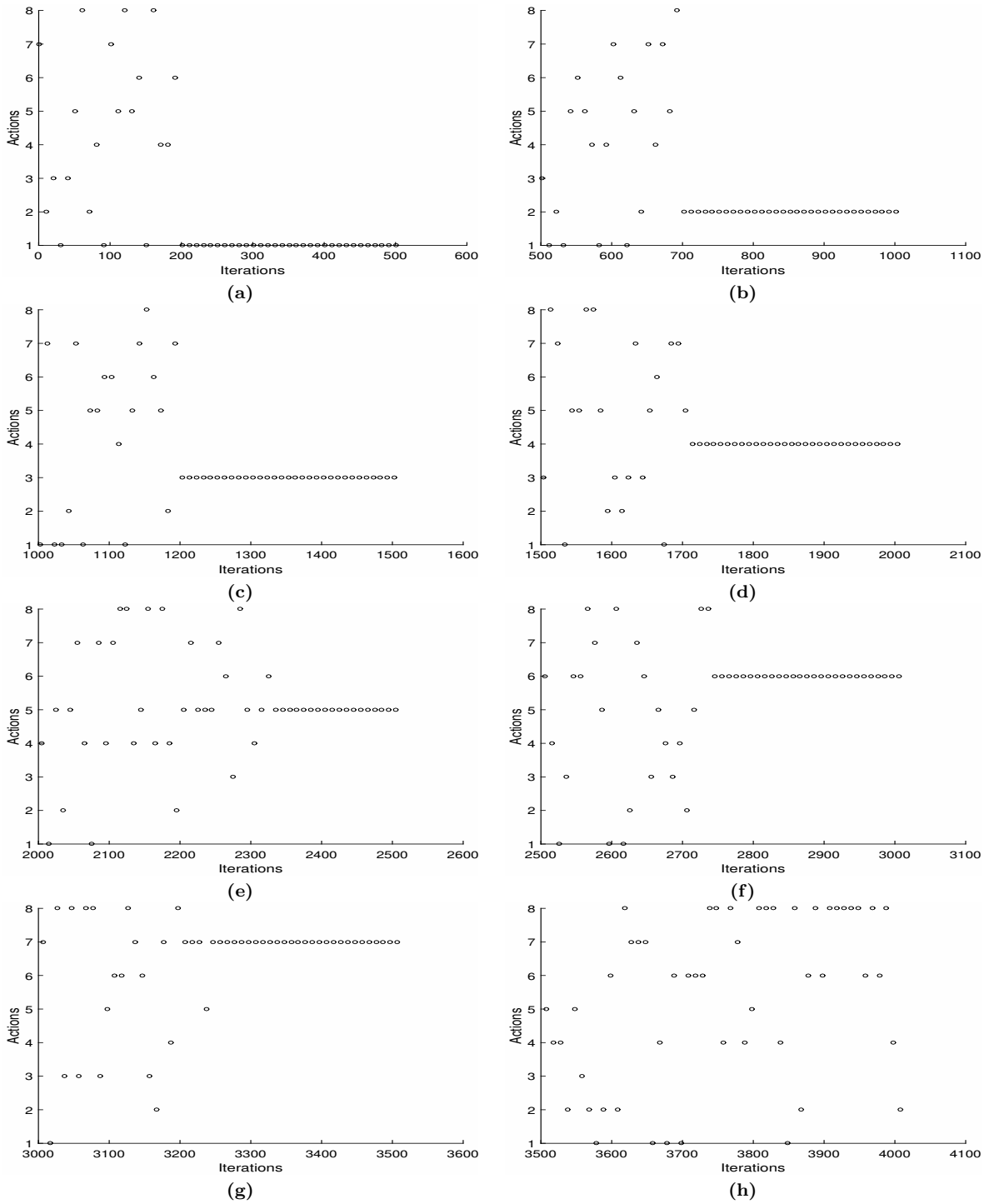


Figure 5.13: Progress plot for input image Fig. 5.1(c) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(c) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

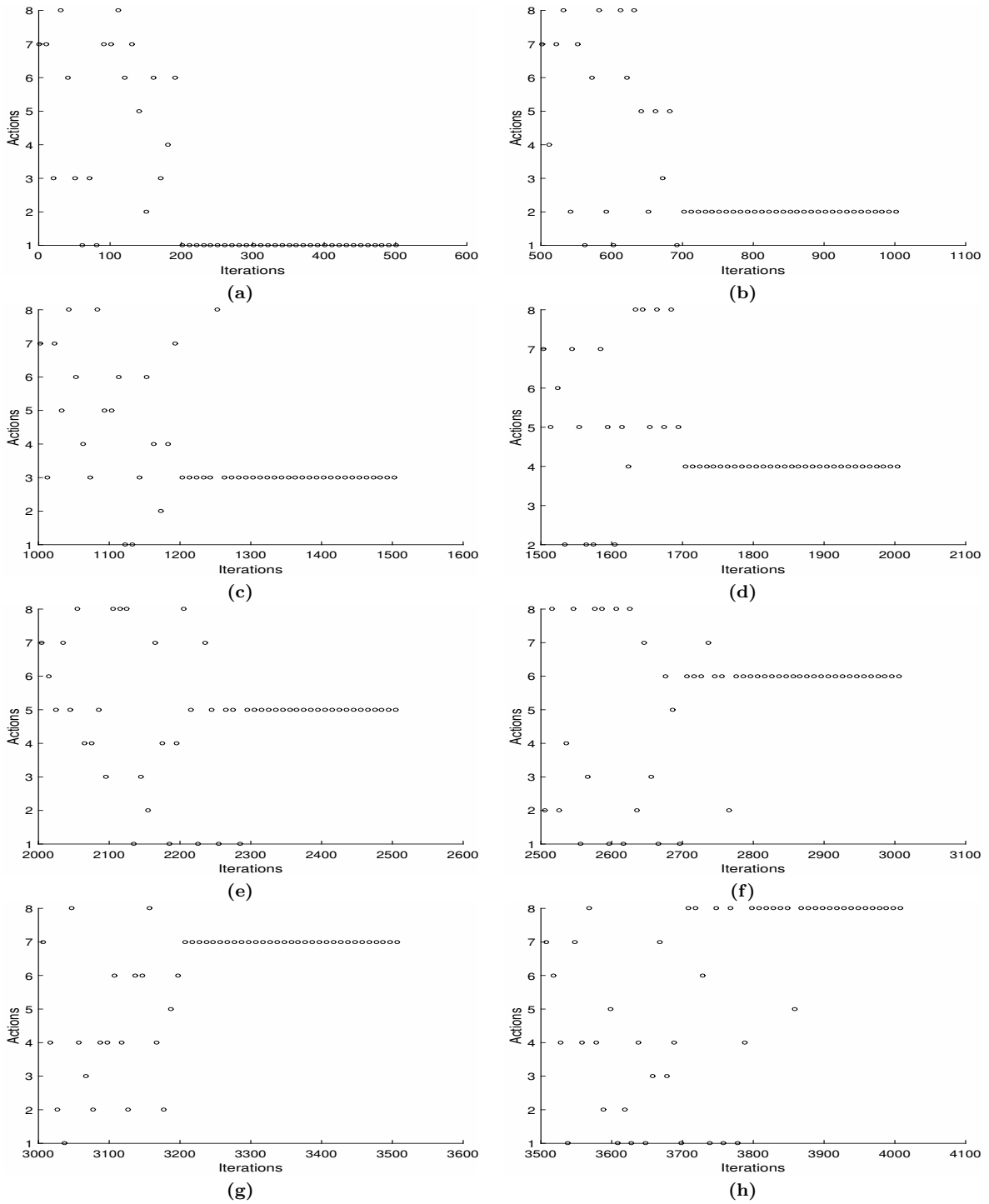


Figure 5.14: Progress plot for input image Fig. 5.1(d) in Scenario 2. The progress graphs shows the actions chosen over training iterations, trained for input image Fig. 5.1(d) for Scenario 2. (a) Training with RYRL initial condition. (b) Training with RYLR initial condition. (c) Training with RPUD initial condition. (d) Training with RPDU initial condition. (e) Training with RRCL initial condition. (f) Training with RRAC initial condition. (g) Training with TLRI initial condition. (h) Training with TLRO initial condition.

5.3 Analysis

The learned R-table becomes quite similar to the ideal R-table when the learning algorithm is executed for a large number of iterations. The algorithm performs very well for images that have high contrast and lots of texture. This is expected as the local motion detectors give smooth optical flow pattern when the image has lot of variation in intensity across the scene. This can be seen in Fig. 5.3 and 5.9, which shows learned R-tables for input images in Fig. 5.1 over Method I and II, respectively. It is quite clear from this result that the algorithm performs best for the image in Fig. 5.1d as it has dense texture and high contrast. The synthetic image (Fig. 5.1a) performs sub-optimally due to sparse texture, while natural images (Fig. 5.1b and c) give better results but not perfect result due to low contrast.

When comparing results over two methods as shown in Scenarios 1 and 2, it can be observed that except for some minor differences both methods yield similar results for all input images. This is expected as the generated states depend more on the position and general direction in which an LPTC responds than subtle variation of axes along which these neurons respond. While matched filter model yields more accurate neural response vis-a-vis natural VS-cells of the fly, it doesn't have much effect on the outcomes of the learning algorithm. It is also interesting to see the behaviour of the agent during the training phase when exposed to different input images. The choice of actions is observed to be random in the first 200 iterations of the training phase with respect to every sensory primitive, as is expected. Subsequently, the learning algorithm converges the choice of action to the desired action for every sensory primitive (Fig. 4.3).

The results show that the proposed agent model and the learning algorithm are able to give promising results for the motion grounding problem in synthetic and

natural scenarios. The agent learns a unique relationship between each of the sensory states and actions. Fig. 5.3 and 5.9 show that those action that are able to maintain and reinforce the sensory state are mapped to that state in the learning process. For example, consider the case where the sensory state is roll optical flow in clockwise sense (RRCL). Now, the action roll in counter clockwise sense (counter clockwise arrow) will generate the same optical flow as observed by the agent. Thus, this action reinforces the current sensory state while other actions would change the state, making the learning algorithm map roll counter clockwise action to generate roll clockwise optical flow (RRCL). A similar relationship can be observed for all other corresponding state-action pairs.

6. DISCUSSION

An interesting question is whether the framework presented in this thesis can be used to design experiments for fly electrophysiology and behavior. These results offer a strong prediction that can be tested in the experimental lab: electrical stimulation of the optical flow detectors in the fly brain would elicit motor behavior that reinforce the detected optical flow. This would also be useful in verifying whether flies actually learn to understand the semantics of their neural spikes generated by optical flow detectors in this fashion. For example, stimulating the RRCL detector will result in counter clockwise rotation.

This framework can also be useful for implementation on real robotic agent to test its workability. This will also present a good understanding of how an agent behaves when actively interacting with real world. Since this approach is proposed to be a basis for developing an agent that understands motion, so it can be built up to add more complex capabilities to the agent.

In future work, I would like to investigate compositional optical flow generated by a combination of multiple optical flows mentioned in this approach: for example, optical flow induced by combined roll and thrust (forward movement), etc. Another aspect related to understanding motion is learning the motion generated by external agents. This is frequently seen in the fly's behavior when it performs complex actions like pursuit of the prey. Processing such motion will give the agent an understanding of the effect of self-motion vis-a-vis another agent's motion. Such motions will have complex optical flow fields.

7. CONCLUSION

In this thesis, I presented an optical flow processing model for a robotic agent that is inspired from the fly’s visual system. This sensorimotor model has been shown to autonomously learn the semantics of internal sensory spikes generated by the motion detectors modeled after the fly’s compound eyes. The agent is able to achieve this by mapping the optical flow input to behaviorally meaningful actions: simple motions like translation, rotation, etc. A good sensorimotor mapping achieved by using the sensory invariance criterion applied to reinforcement learning shows that the agent is able to autonomously learn the meaning of internal spikes in terms of actions without having any direct access to the outside stimuli. I have shown results of this for various synthetic and natural scenes giving best mapping for scenes with high contrast and dense texture.

The ability of a robotic agent to autonomously ground optical flow information can act as a basis for learning complex information using developmental learning. The approach used in this thesis can be useful for further developing the capability of such an agent. Moreover, I expect this computational study to help better understand how the real fly visual system decodes its own spikes.

REFERENCES

- [1] Y. Choe and N. H. Smith, “Motion-based autonomous grounding: Inferring external world properties from encoded internal sensory states alone,” in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, p. 936, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [2] P. Rieke, D. Warland, R. d. Ruyter van Steveninck, W. Bialek, and D. Reich, “Spikes: Exploring the neural code,” *Network-Computation in Neural Systems*, vol. 8, no. 3, p. 353, 1997.
- [3] F. Raudies, “Optic flow,” *Scholarpedia*, vol. 8, no. 7, p. 30724, 2013. revision 149632.
- [4] G. K. Taylor and H. G. Krapp, “Sensory systems and flight stability: What do insects measure and why?,” *Advances in Insect Physiology*, vol. 34, pp. 231–316, 2007.
- [5] J. P. Kumar, “Signalling pathways in drosophila and vertebrate retinal development,” *Nature Reviews Genetics*, vol. 2, no. 11, pp. 846–857, 2001.
- [6] S. Duke-Elder, *System of Ophthalmology: The Eye in Evolution*, vol. 1. Mosby, 1958.
- [7] K. Hausen, “Motion sensitive interneurons in the optomotor system of the fly,” *Biological Cybernetics*, vol. 45, no. 2, pp. 143–156, 1982.
- [8] A. Borst and M. Egelhaaf, “Principles of visual motion detection,” *Trends in Neurosciences*, vol. 12, no. 8, pp. 297–306, 1989.

- [9] M. Egelhaaf, N. Boeddeker, R. Kern, R. Kurtz, and J. P. Lindemann, “Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action,” *Frontiers in Neural Circuits*, vol. 6, 2012.
- [10] H. G. Krapp, B. Hengstenberg, and R. Hengstenberg, “Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly,” *Journal of Neurophysiology*, vol. 79, no. 4, pp. 1902–1917, 1998.
- [11] H. Cuntz, J. Haag, and A. Borst, “Modelling the cellular mechanisms of fly optic flow processing,” in *The Computing Dendrite*, pp. 259–275, Springer, 2014.
- [12] M. Egelhaaf, R. Kern, H. G. Krapp, J. Kretzberg, R. Kurtz, and A.-K. Warzecha, “Neural encoding of behaviourally relevant visual-motion information in the fly,” *Trends in Neurosciences*, vol. 25, no. 2, pp. 96–102, 2002.
- [13] J. Lindemann, R. Kern, J. van Hateren, H. Ritter, and M. Egelhaaf, “On the computations analyzing natural optic flow: Quantitative model analysis of the blowfly motion vision pathway,” *The Journal of Neuroscience*, vol. 25, no. 27, pp. 6435–6448, 2005.
- [14] H. G. Meyer, J. P. Lindemann, and M. Egelhaaf, “Pattern-dependent response modulations in motion-sensitive visual interneurons—a model study,” *PLoS One*, vol. 6, no. 7, p. e21488, 2011.
- [15] H. Eichner, M. Joesch, B. Schnell, D. F. Reiff, and A. Borst, “Internal structure of the fly elementary motion detector,” *Neuron*, vol. 70, no. 6, pp. 1155–1164, 2011.

- [16] W. Reichardt and W. Rosenblith, “Autocorrelation, a principle for evaluation of sensory information by the central nervous system,” in *Symposium on Principles of Sensory Communication 1959*, pp. 303–317, MIT Press, 1961.
- [17] W. Reichardt, “Evaluation of optical motion information by movement detectors,” *Journal of Comparative Physiology A*, vol. 161, no. 4, pp. 533–547, 1987.
- [18] S. Harnad, “The symbol grounding problem,” *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 335–346, 1990.
- [19] Y. Choe, H.-F. Yang, and D. C.-Y. Eng, “Autonomous learning of the semantics of internal sensory states based on motor exploration,” *International Journal of Humanoid Robotics*, vol. 4, no. 02, pp. 211–243, 2007.
- [20] F. A. Miles and J. Wallman, *Visual Motion and its role in the Stabilization of Gaze*, vol. 5. Elsevier Science Ltd, 1993.
- [21] M. Lappe, “Computational mechanisms for optic flow analysis in primate cortex,” *International Review of Neurobiology*, vol. 44, pp. 235–268, 2000.
- [22] J. J. Gibson, *The Perception of the Visual World.*, vol. xii. Houghton Mifflin, 1950.
- [23] K. Nakayama and J. Loomis, “Optical velocity patterns, velocity-sensitive neurons, and space perception: a hypothesis,” *Perception*, vol. 3, no. 1, pp. 63–80, 1974.
- [24] J. J. Koenderink and A. J. van Doorn, “Facts on optic flow,” *Biological Cybernetics*, vol. 56, no. 4, pp. 247–254, 1987.
- [25] B. D. Lucas, T. Kanade, *et al.*, “An iterative image registration technique with an application to stereo vision.,” in *IJCAI*, vol. 81, pp. 674–679, 1981.

- [26] B. K. Horn and B. G. Schunck, "Determining optical flow," in *1981 Technical Symposium East*, pp. 319–331, International Society for Optics and Photonics, 1981.
- [27] H. G. Krapp, "Neuronal matched filters for optic flow processing in flying insects," *International Review of Neurobiology*, vol. 44, pp. 93–120, 2000.
- [28] B. Hassenstein and W. Reichardt, "Der schluß von reiz-reaktions-funktionen auf systemstrukturen," *Zeitschrift für Naturforschung B*, vol. 8, no. 9, pp. 518–524, 1953.
- [29] N. Franceschini, A. Riehle, and A. Le Nestour, "Directionally selective motion detection by insect neurons," in *Facets of Vision*, pp. 360–390, Springer, 1989.
- [30] A. Borst and M. Egelhaaf, "Detecting visual motion: Theory and models," *Visual Motion and its Role in the Stabilization of Gaze*, pp. 3–27, 1993.
- [31] K. Hausen, "Decoding of retinal image flow in insects.," *Reviews of Oculomotor Research*, vol. 5, pp. 203–235, 1992.
- [32] H. G. Krapp and R. Hengstenberg, "A fast stimulus procedure to determine local receptive field properties of motion-sensitive visual interneurons," *Vision Research*, vol. 37, no. 2, pp. 225–234.