GRIDBOT: TOWARDS A NEUROINSPIRED NAVIGATION SYSTEM FOR ROBOT PLANNING

BY GUANGZHI TANG

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Computer Science

Written under the direction of Konstantinos P. Michmizos and approved by

New Brunswick, New Jersey

May, 2017

ABSTRACT OF THE THESIS

Gridbot: Towards a neuroinspired navigation system for robot planning

by Guangzhi Tang Thesis Director: Konstantinos P. Michmizos

The ability to orient in an unknown, fast-changing, environment is an unmet challenge for robots but a seamlessly solved problem for the primate brain. This thesis describes the first steps in developing a neuro-inspired bottom-up model of the brains navigation system to make a mobile robot localize itself, map its surrounding and plan its trajectory. Our model employs neural spikes to encode and process information in real-time. Despite a multitude of Nobel-winning studies that have revealed neurons specializing in self-navigation, such as place, grid, border and head direction cells, their interconnectivity remains elusive. Therefore, any model employing these neurons needs to make quite a lot of extrapolations to fill in the gaps of knowledge. The main challenge was to design a real-time spiking neural network that can compensate for the hardware limitations as well as its own intrinsic imperfections and work in real conditions. To design the first component of our model, the head direction cell layer, we employed mechanisms based on self-organizing and self-sustaining neural activity, or attractor dynamics, resembling those originally proposed in Hebbs cell assembly theory. The information to be maintained and updated was a continuous variable, or continuous attractor, where a 1D continuum of cell assemblies represented head direction. In theory, our network should give rise to a self-sustained hill of excitation - the attractor. In practice, due to non-ideal speed sensors and the intrinsic spike variability of the spiking network, it was impossible to sustain a correct approximation of the head direction using just this scheme. To correct this, we introduced a spike-based Bayesian inference layer of leakyintegrate-and-fire models of neurons, that combined feedforward (vision) and recursive (kinesthetic) inputs. We show how such a layer can approximate the posterior probability of the preferred state encoded in the spiking probability by adding the logarithms of the simulated dendritic currents, which is a reasonable approximation of the nonlinear dendritic activity. We show that our model accurately estimated head direction and further extend it to include a dynamic network of border cells that can learn to map the observed environment through simulating synaptic plasticity. Solving the localization problem and creating a cognitive map of the surroundings, our thesis paves the way for tackling robot planning through imitating brain structure, its principles and its performance.

Acknowledgements

I would first like to thank my thesis advisor Professor Konstantinos Michmizos of the Computational Brain Lab at Computer Science Department of Rutgers University. His door is always open for me, and we have lots of discussions nearly everyday, especially the hours talks on Friday afternoons after group meeting. I would also like to thank to all team members in the Computational Brain Lab, especially Vladimir, Leo, and Praveen. Without them the validation thesis could not have been successfully conducted.

Table of Contents

Abstract
Acknowledgements in
List of Tables
List of Figures
1. Introduction
1.1. Problem statement
1.2. Background research
1.3. Description of the remaining chapters
2. Computational Spatial Neuron Models
2.1. Continue attractor neural network model for head direction cell
2.2. Boundary vector cell model
2.3. Probabilistic learning model for grid cell
3. Head direction cells model employing Bayesian inference
3.1. Why we built a accurate head direction cell model?
3.2. Bayesian inference in spiking neurons
3.3. Vision cue for head direction cells
3.4. Head direction cells employing Bayesian inference
3.5. Model implementation in ROS 10
3.6. Experiments and results
4. Dynamic border cell mapping model
4.1. Why we built a dynamic border cell mapping model?

4.2.	Egocentric border cells	23
4.3.	Allocentric border cells	25
4.4.	Mapping the environment employing synaptic plasticity	26
4.5.	Model implementation in ROS	30
4.6.	Simulation and results	30
5. Co	nclusion	34
Biblio	graphy	36

List of Tables

3.1.	Parameters in simulation experiment for HD cell model	19
4.1.	Parameters in simulation experiment for Border Cell model	31

List of Figures

3.1.	HMM implemented in spiking neural networks	11
3.2.	Structure of the HD cell model and 3 different kinds of synaptic connections $% \left(\frac{1}{2} \right) = 0$	13
3.3.	Nodes and topics graph of HD cell model from ROS $\ldots \ldots \ldots \ldots$	17
3.4.	Simulation environment in Gazebo simulator for HD cell model experiments	18
3.5.	Tuning curve for 2 simulated HD cells	20
3.6.	A partial raster map for HD cells layer neurons	20
3.7.	result from a long experiment where the robot rotates with angular ve-	
	locity of 10 deg/s	21
4.1.	Transform laser scan signal into spiking of neurons in egocentric border	
	cell layer	24
4.2.	Transform egocentric border information into allocentric border informa-	
	tion using HD cells	26
4.3.	Learning environmental map using synaptic plasticity, each neuron in	
	allocentric border cell layer only connect with its surrounding neurons $% \mathcal{A}$.	28
4.4.	Nodes and topics graph of border cell model from ROS	30
4.5.	Experiment environments for border cell model	31
4.6.	Experimental map generated in 3 different environments, colors in the	
	map represents strength of border cell activity, x-axis represents pre-	
	ferred directions and y-axis represents preferred distances. a) Simulator	
	environment of room. b) Simulator environment of separate objects. c)	
	Real world environment in the lab	32

Chapter 1

Introduction

1.1 Problem statement

Moving and path planning in dynamic environments are crucial tasks for the survival of mobile agents. To design any efficient planning strategy, one needs to be fully aware of the position and the surrounding environment. In robotics, this coupled problem is defined as localization and mapping problem. Such problems have long been at the foundation of mobile robotics [1].

Interestingly, localization and mapping are important tasks for humans and animals. Extensive research has been done on the neural signature of the ability of the brain to orient itself. These studies have lead to a number of newly discovered cells, such as place cell [2], grid cell [3], border cell [4], head direction cell (HD cell) [5], and many other special function neurons [6][7].

Although a unified theory of how such a spatial system in the brain works, a series of computational models have supported, or triggered, experimental findings. These models try to fill in the gaps between the already known neurons and propose specific roles for each one of the neural parts. Place cells in the hippocampus are related to path integration and planning [8]. Grid cells are related to speed integration and localization [9]. Border cells are representations of environment information [10]. Head direction cells give orientation information to the spatial system [11].

producing accurate results from a noisy system that is driven by noisy sensors is a main challenge for localization and mapping. biological networks of neurons are also noisy systems and they can be influenced by many factors. Sensors are noisy in the real world. In mobile robot, motion sensors and depth cameras have huge errors compared to the ground truth. In the brain, information come from eyes or vestibular system are full of noises and errors [12][13]. These noises and errors make it impossible to trust any sensory system separately. In the brain, information coming from the eyes or the vestibular system is noisy and error-prone [12][13]. The presence of noise and other errors makes it impossible to trust any sensory system separately. The brain combines information from different sensory systems. An interesting recent finding from an experimental and computational study shows that neural areas can achieve Bayesian optimal results [14]. In robotics, different sensors are also used to generate accurate results for future computations in a similar fashion, such as in the simultaneous localization and mapping (SLAM) algorithms [1].

Mapping the environment is an important task for both the mapping and the localization problem. Although multiple neurons have been related to the spatial system of the brain, there is currently no neurophysiologically plausible model that explains how the environmental map can be represented in the brain. One of the challenges for building an environmental map is how we should maintain a stable and dynamic set of memories that relies on sensory input. Sensory observations of the environment is very limited at a single time point: The human field of view only covers a slightly over 180-degrees horizontally. Nevertheless, we need to memorize landmarks in multiple directions in order to build a map for the entire environment. Employing synaptic plasticity, in the form of long-term potentiation (LTP) and long-term depression (LTD) [15], seems to be an appropriate first step to form these memories. The map of the environment can be stored in the weights of the synapses and change dynamically using local, Hebbian-based, synaptic plasticity rules.

This thesis makes the first and most important steps towards building the Gridbot system, a neuro-inspired navigation system that helps robot to localize itself, map its surrounding and plan its trajectory. By mimicking the current findings of brain spatial system, we designed a real-time, robust spiking neural network that can compensate for the hardware limitations as well as its own intrinsic imperfections. Specifically, we built a neuro-inspired environmental mapping model on mobile robot. This neuromimetic robotic system follows a bottom-up approach and relies on neural spiking to self-map autonomously and on real-time. The construction of the model was done in two steps. First we built an HD cell layer employing Bayesian inference that can provide an accurate head direction representation using the real-time spiking of neurons. Second we built a dynamic border cell network that learns the map in different environments using synaptic plasticity. Experiments both in simulator and real world environments show the model successfully uses spiking neurons to generate a map of the environment. As a result, the model provides a powerful assumption of how border cell network works intrinsically, and may provide a good direction for neuroscientists to uncover the biological truth behind neural connectomics for the brain navigational system.

1.2 Background research

Spatial system of the brain has been studied in the past decades, and lots of related neurons has been found. In 1971, place cell was found in the hippocampus of moving rats [2]. Place cell becomes active when an animal enters a particular place field in the environment. In 1984, head direction cell (HD cell) was discovered in the dorsal presubiculum of rats [5]. HD cell raise its firing rate when animal's head is reaching a preferred direction. In 2005, grid cell was found in the medial entorhinal cortex [3]. Different from place cell, grid cell reacts to multiple place fields in the environment, and all these place fields forms a triangular lattice. It is believed by researchers that grid cell gives a metric representation to the spatial system [16]. In 2009, border cell was discovered in the hippocampal formation [4]. Border cell is a kind of neuron related to the environmental map in the brain. Border cell becomes active when there is a border at preferred direction and distance.

Since the interconnections between these neurons, and the mechanisms under their activities are still unknown, multiple computational models have been developed trying to give assumptions of these neurons. The most popular model is the continue attractor neural network (CANN) model. First developed for HD cells [11], CANN model has also been generalized onto place cells [8] and grid cells [9]. Experiment results show grid cell generates weeks after birth [17], a model generalize CANN model with synaptic plasticity has been proposed to give an assumption of this process [18].

There are other models for neurons in brain spatial system as well. Boundary vector cell (BVC) model not only gives an assumption of border cell, but also generates place cell activity using borders of the room-like environment [19]. There is also the oscillatory interference model for grid cell [20], trying to simulate the activity of grid cell in a oscillatory point of view. Different from CANN, all these models are more difficult to implement using spiking neurons.

Movements of mobile robots share a lot similarity with moving mammals, like rats or human. They share similar inputs and goals. Because of this, there are also many computational models for neurons in brain spatial system inspired by robotic stateof-the-art methods. The probabilistic learning model for grid cell [10] is inspired by fastSLAM algorithm [21] in robotics. By combining the visual input and self-motion input, the probabilistic learning model generates accurate location representation from grid cells.

1.3 Description of the remaining chapters

There are 4 remaining chapters in this thesis. Chapter 2 briefly explains the computational neuron models that inspired our approach. These models are mainly focus on HD cell and border cell. We propose our model in Chapter 3 and Chapter 4. Chapter 3 explains the intrinsic structure of HD cell model employing Bayesian inference. Chapter 4 discovers the use of depth camera and HD cell for generating border cell allocentric activity, and the use of synaptic plasticity for forming memory between border cell interconnections. Chapter 5 gives the conclusion of our work and the future generalization.

Chapter 2

Computational Spatial Neuron Models

2.1 Continue attractor neural network model for head direction cell

Continue attractor neural network (CANN) model has been used to simulate the behavior of multiple neurons in spatial system of brain [22]. CANN model generalize the concept of attractor model, a kind of computational model for discrete memories. Attractor model such as Hopfield network [23] stores memory in self-organizing and self-sustaining neural activities, or attractor dynamics. It is very hard for the model to transfer from memory to memory since each memory is in its own lowest energy attractor.

The information to be maintained and undated in CANN model is not a set of discrete items, head direction or body position are changing rapidly through time. In CANN model, attractors no longer lie separately on a surface as in Hopfield network. They become continuous variables, or continue attractors, and lie continuously on a manifold from high dimension space. For example, continue attractors for HD cells lie on a 1D circle manifold [24] and continue attractor for grid cells lie on a 2D twist torus manifold [25]. Though sacrifice part of the stability of memory, continue attractor makes memory a lot easier to transfer.

CANN model for HD cells uses a group of neurons to sample a 1D circle manifold of head direction. Each neuron has a unique preferred head direction, and all the neurons are uniformly distributed on the circle. Each neuron in the model has synapses with all other neurons in the network. Neurons in close association excite each other with the amount of excitation being proportional to the difference of preferred directions between neurons. Distant neurons are connected with inhibitory connections. Since the weights each neuron connected to all other neurons are symmetrical, an stationary attractor representing head direction can be formed and sustained under no external influences. When there is a external stimuli, such as speed, the symmetry of the network is temporarily distorted by biasing the activities of neurons. This distortion causes the network trying to get another stable state, therefore drives the attractor to move.

Many researches have been done on CANN model for HD cells. Most of the researches are focusing on rate model but not spiking model of neuron. CANN model has also been implemented onto mobile robot [26][27]. However, due to non-ideal speed sensors and the intrinsic spike variability of spiking neurons, it was impossible to sustain a correct approximation of the head direction using spiking CANN model in practice.

2.2 Boundary vector cell model

A good example to show how computational models help the understanding of biological neuron systems in the brain is the boundary vector cell (BVC) model proposed in 1990s [28]. Experimental results show place cell activity is highly related to the distances between preferred place field and borders of the environment. Therefor, BVC model is proposed assuming a predicted boundary vector cell contributes to the firing of place cell. Amazingly border cell is found in 2009 [4] having very similar activity as the predicted boundary vector cell.

$$df_t = g_t(d,\theta)d\theta \tag{2.1}$$

$$g_t(d,\theta) \propto \frac{exp[(d-r_i)^2 \ 2\sigma^2(r_i)]}{\sqrt{2\pi\sigma^2(r_i)}} \times \frac{exp[(\theta-\phi_i)^2 \ 2\sigma^2(\phi_i)]}{\sqrt{2\pi\sigma^2(\phi_i)}}$$
 (2.2)

Though BVC model made a huge contribution to the finding of border cell, the model itself is actually a very simple one without any input to border cell. The model assumes a rat in a room already know all positions of borders surrounding it. For instance there is a border cell in the model with preferred direction ϕ_i and preferred distance r_i . The updating rule for border cell in BVC model is shown in Equation 2.1

and 2.2. Since positions of all borders at each direction are already known, border cell activity is simulated by a integral of a joint Gaussian distribution for all directions.

BVC model doesn't give a biologically feasible explanation of how border cell activity is generated. It neglects the transformation between egocentric information from vision and allocentric information of the environmental map. It also failed to explain how observed borders are memorize in border cell network. All these questions are waiting to be answered by more biologically feasible computational models for border cells.

2.3 Probabilistic learning model for grid cell

It is very hard for computational models to satisfy both biologically feasible and working in real world simulation. Most real world working models of spatial neurons have to get inspirations from robotics. The probabilistic learning model [10] try to explain simultaneous learning between information from border cell and grid cell by adapting the framework from fastSLAM algorithm [21]. It uses rate model to simulate border cell and grid cell, and proposes a spatial information fusion model (SIFM) algorithm to generate accurate border cell and grid cell activities.

Different from other computational models for grid cell, the model defines position represented by grid cell using oscillatory grid code. Each grid code contains multiple oscillatory grid cell rate model [20] with different grid scales and random grid phase, and a head direction representation. Therefore each grid code in the model represent one unique position in environment.

Two kinds of border cells are simulated in the model. Sensory border cells react to sensory inputs of border positions using mechanism similar to BVC model. Predictive border cells generate associative maps of the environment for each position. Since each grid code is a sampled position of the environment, different groups of predictive border cells are connected to each one of the grid code separated representing the possible map of positions.

Associative maps represented by predictive border cells are learned using corresponding grid codes and sensory border cells information. A uniform rectangular lattice in the environment is used as a base to update the associative weight between grid code and predictive border cells. Each time-step, the associative weight is updated by the joint activities between grid code and sensory border cells. Eventually, associative map of one grid code is generated by a weight summation of elements in the lattice.

Limitation for the probabilistic learning model is very obvious. First it uses very high level models for simulating neurons. Both grid code and border cell in the model are more like transforming SLAM concepts into biological terms. The methods based on grid code, such as directly sampling and associative learning, are almost impossible to implement on lower level neuron models, like spiking neuron model. However, the idea of the model is very fascinating. It treats grid cells as a sample of environment, and each sample has its own predicted map. This idea inspired our approach on border cell.

Chapter 3

Head direction cells model employing Bayesian inference

3.1 Why we built a accurate head direction cell model?

Direction is the most basic and fundamental element in navigation. It contributes to both movement and observation, two of the most important tasks in navigation. Head direction of moving agent combines both moving direction and visual direction, since it moves and observes in the direction its head points at most of the time. Head direction cells (HD cells) have been found in the brain of most mammals. Experimental studies show HD cells give accurate representations of direction. Though visual and vestibular influences on HD cells have been observed [29], there is no neurophysiological explanation on how the head direction representations on HD cells are formed [30].

The most used spiking neural network model for HD cells in computational neuroscience is the CANN model. The classic HD cell CANN model only uses self-motion input to drive the activity of HD cells, the head direction represented by the model results with a drift error when big noises are included in self-motion input. Other HD cell CANN models try to solve this problem by using visual input. However, mammalian visual system provides very noisy visual information. No HD cell model has been proposed to eliminate both self-motion and visual errors simultaneously with a spiking neuron network.

An accurate HD cell model is essential to our Gridbot spiking neuron system. Most importantly, the model gives an assumption of the structure of HD cells and related neurons. It has the potential to point the way for neuroscientists for better understanding of HD cell system in the brain. Therefore, we propose a spiking HD cell model employing Bayesian inference generating accurate head direction representation.

3.2 Bayesian inference in spiking neurons

Bayesian inference is a method to update hypothesis using extra evidence base on Bayes' theorem. The formal definition of Bayesian inference is shown in equation 4.1. P(H|E) is the posterior probability, it gives the probability of hypothesis H after new evidence E has been observed. P(H) is the prior probability, it estimates the probability of hypothesis H before any new evidence is observed. P(E|H) is the likelihood, it shows the probability of observing evidence E with the given hypothesis H. P(E) is the probability of observing evidence E, and since it stays the same for any hypothesis, it can be ignored in Bayesian inference computation.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \propto P(E|H)P(H)$$
(3.1)

Bayes filter is a method using Bayesian inference recursively at each time step to estimate the probability of hypothesis of state, or belief of state. Here the belief of state represents a distribution of head direction of mobile creature or robot. Bayes filter can be represented by a hidden Markov model (HMM). In HMM belief of state is propagated through time steps using equation 4.2. At each time step k, belief of state x_k is represented by posterior probability $P(x_k|u_1, ..., u_k, z_1, ..., z_k)$, u_k is the transition probability from time step k-1 to time step k, z_k is the observation of environment at time step k. Belief propagation through HMM is generated from Bayesian inference base on Markov assumption between time steps.

$$Bel(x_k) = \eta P(z_k | x_k) \int P(x_k | u_k, x_{k-1}) Bel(x_{k-1}) dx_{k-1}$$
(3.2)

Experiments shows that the brain performs Bayesian inference for variety of tasks, such as cue integration and decision making [31]. Multiple models have been proposed to perform Bayesian inference on spiking neurons [14][32]. Since spiking neurons can perform Bayesian inference, they can also perform Bayes filter over time.

HMM implemented in spiking neural networks [33] is shown in Figure 3.1. Neurons in Bayesian inference layer receive inputs from both evidence likelihood layer and state belief layer, every preset time step state belief layer will be updated by a bursting input



Figure 3.1: HMM implemented in spiking neural networks

from Bayesian inference layer. Updating rule for neurons in Bayesian inference layer using leaky-integrate-and-fire (LIF) neuron [34] is shown in equation 4.3. Equation 4.4 and 4.5 show belief propagation through HMM in Equation 4.2 can be implemented using spiking neural network if it is performed in the logarithm domain [35]. By using non-linear dendritic functions to approximate the logarithm function, neuron activities in Bayesian inference layer will eventually proportional to logarithm of belief of state through time.

$$\tau \frac{d}{dt} v_i = -v_i + R(f(\sum_j w_{ij}^{(z)} I_j^{(z)}(t)) + g(\sum_j w_{ij}^{(x)} I_j^{(x)}(t)))$$
(3.3)

$$f(\sum_{j} w_{ij}^{(z)} I_{j}^{(z)}(t)) = \log P(z_{k} | x_{k})$$
(3.4)

$$g(\sum_{j} w_{ij}^{(x)} I_{j}^{(x)}(t)) = \log \int P(x_{k} | u_{k}, x_{k-1}) Bel(x_{k-1}) dx_{k-1}$$
(3.5)

Synaptic current inputs from evidence likelihood layer and state belief layer generate a new distribution on Bayesian inference layer by using equation 4.3. This distribution has the same mean value with the Bayesian inference of two Gaussian distribution from the pre-synaptic layers, and variance of the distribution is controlled by the connection between pre and post-synaptic layers. Though current input to Bayesian inference is performed in the logarithm domain, dendrites of neurons in the layer make the distribution expressed by the layer looks very similar to a Gaussian distribution.

3.3 Vision cue for head direction cells

Experimental studies show visual information contributes to the accuracy of head direction representation in HD cells [29]. The absence of visual input to the biological HD cells introduces a gradual drift due to an error accumulation. This phenomenon can be observed easily in dark environments, with the sense of orientation weaken dramatically due to lack of visual information. However, visual information can't provide the ground truth of head direction, since it is also a noisy signal. In probability model, visual information is the observation of environment when unknown ground truth head direction is reached. It contributes as a landmark likelihood to estimate the ground truth.

Landmark likelihood is hard to compute in real world problems. The easily way for computing the likelihood is to give a advance knowledge of the environment to the robot. Suppose the robot knows the estimated position of a certain landmark, then we can compute a distribution representing the probability of observing the landmark at different directions, this distribution is the landmark likelihood.

There are also brain-mimicking ways to compute the landmark likelihood. One possible way is to use border cells to form a estimated map of the environment for likelihood computation. The border cells map can be learned using visual sensory inputs and synaptic plasticity. After we get the map of the environment, we can use the visual observation to compare with the estimated map. A group of similarity indexes can be generated based on guesses of the head direction performing the observation, these similarity indexes have equivalent meanings with the landmark likelihood distribution.

3.4 Head direction cells employing Bayesian inference

There are three components of our spiking neural network model for head direction cells. The core component is a CANN model for HD cell with 1 HD cells layer and 2 HD transition layers [11]. Another two components are the landmark likelihood layer, and the Bayesian inference layer. Each layer in the model has 360 LIF neurons, the resolution for 1 head direction cell is 1 degrees. Synaptic currents are used between neuron connections, the contribution of spikes to synaptic current decay exponentially with time.



Figure 3.2: Structure of the HD cell model and 3 different kinds of synaptic connections

Synaptic connections between different layers are shown in Figure 3.2. Neurons in HD transition layers receive current inputs directly from angular speed sensor. Attractor bump on HD cells layer represents the distribution of head direction, the bump is drove by HD transition layers and corrected by Bayesian inference layer in every 500 ms time step. Overall HD cells layer receives information from angular speed sensor and visual sensor, but not directly, since none of the input from two sensors is fully trustful. Bayesian inference layer combines cues from two sources and generates a more accurate distribution.

Different kinds of synaptic connections between pre and post-synaptic layers are used in the model. The most used synaptic connection in the model is the one-on-one excitatory synapse between layers, this connection transfers information between layers plainly without any reinforcement.

Another synaptic connection is the center excitatory global inhibitory synapse. Every neuron in pre-synaptic layer has a corresponding neuron in post-synaptic layer. By using this connection, pre-synaptic neuron will has excitatory synapses with neurons near its corresponding post-synaptic neuron, and has inhibitory synapses with all other neurons far away. Weights of the synapses are determined by the distance between neurons. Distance between neurons here is defined by the semantic meaning represented by the layer, here it represents the distance of angles represented by neurons. Single bump distributions, like Gaussian distribution or other similar distributions, can be more accurately transfered between layers with this connection.

The last kind of synaptic connection used in the model is the shifted center excitatory global inhibitory synapse, it is only used inside the CANN component of the model. Different from the previous connection, this connection shifts the post-synaptic neurons when connecting the synapses. The shift in synaptic connection results a shift of mean value between distributions in pre and post-synaptic layer.

Equation (1) shows the voltage updating rule for neurons in HD cells layer. Neurons in the layer receive current inputs from three sources. First is the recurrent synapses within the layer. Second is the HD transition current from 1 of the 2 HD transition layers activates attractor's rotation. Third is the bursting correction current from Bayesian inference layer.

$$\tau \frac{d}{dt} v_i^h = -v_i^h + R(\sum_j w_{ij}^{hh} I_j^h + \sum_k w_{ik}^{rh} I_k^r + \sum_m w_{im}^{bb} I_m^b)$$
(3.6)

Each neuron in HD cells layer represents a distinct direction of head. Recurrent synaptic weight follows a Gaussian distribution of distance between pre- and postsynaptic neurons' head directions subtract by a positive constant value. Nearby excitatory and global inhibitory synapses for all neurons in the layer generate a stable attractor state.

HD cells layer receives self-motion information from 2 HD transition layers, clockwise HD transition neuron layer and counter clockwise HD transition neuron layer. Each HD transition neuron has a corresponding HD cell. As shown in equation 4.7, HD transition neuron receives current inputs from angular speed sensor and corresponding HD cell. Neurons in clockwise and counter clockwise layers are identical. The difference between these layers is the weight direction of the synapses from rotation neurons to head direction cells. Clockwise layer's weights perform clockwise direction exponential decay starting at the neuron next to the corresponding head direction cell, and vice versa for counter clockwise layer. Rotation synaptic currents cause a distortion of the attractor state, and make it rotate in one direction.

$$\tau \frac{d}{dt} v_i^r = -v_i^r + R(\alpha^{ar} I_i^a + \beta^{hr} I_i^h)$$
(3.7)

Landmark likelihood layer gives another distribution of head direction generates by environmental observation. As we previously covered, firing pattern of this layer can be generated by different mechanisms. Visual information from landmarks or environmental information from border cells can both contribute to activities of this layer. In our model, the input of this layer is simplified. We add time independent noises and errors onto ground truth direction of the robot, and use it to form the landmark likelihood distribution.

Equation 4.8 shows the voltage updating rule for neurons in Bayesian inference layer. Bayesian inference layer uses the concept of Bayes filter to estimate head direction. Bayes filter is a general algorithm to estimate distribution of hidden state in HMM. It first computes a prior distribution p(x) based on former estimations and transition functions, then uses Bayesian inference and observation likelihood p(z-x) to estimate hidden state's distribution. Bayes filter can't directly be implemented for real world problem, since it is hard to find close forms for prior computation and observation Bayesian inference.

$$\tau \frac{d}{dt} v_i^b = -v_i^b + R(\log(\alpha^{vb} I_i^v) + \log(\beta^{hb} I_i^h))$$
(3.8)

Here HD cells layer and 2 HD transition layers generate prior distribution for head direction. Landmark likelihood layer gives observation of the environment. Neurons in Bayesian inference layer receive current inputs from HD cells layer and landmark likelihood layer. It uses dendritic non-linearity to approximate logarithm of current inputs [36]. Neuron updating voltage using equation (3) will has a tuning curve like Bayesian inference of HD cells layer distribution and landmark likelihood layer distribution.

3.5 Model implementation in ROS

We implement our model in Robot Operating System(ROS) [37], the most popular robotic framework. ROS is a distributed and modular framework very similar to the brain. As we know, neuron is the basic computing unit in the brain, it communicates with other neurons using synapses. Biologically all neurons are paralleled and compute separately, these make the brain performs fast computation using slow biological network. We built a similar network in ROS using paralleled nodes and topics.

Same with neuron to the brain, node is the basic computing unit in ROS. Nodes are packaged into separate threads, and they can be parallel computed using a multithreads system. In ROS, nodes communicates with each other using messages in topics. A perfect simulation for biological neural network using ROS is to use one single node to represent one neuron, use topics to represent synapses between neurons, and use messages to represent neurotransmitters in synapses. However, since we need to build a real-time system to adapt to the dynamic change of the environment, abstraction and optimization have to be made.

Instead of using one node to represent one single neuron, we use it to represent a layer of neurons. Topics between nodes represent synapses between neural layers, they can be recurrent synapses or feed-forward synapses. Synaptic weights are stored in matrices and defined in each node. At each time step (10 milliseconds), neurons in one node are updated using Euler method [38] simultaneous using currents vector from dendrites. Two kinds of messages are transmitted in topics, spike message and current message. A spike message is published each time step with a boolean array showing if the neurons are spiking or not at the current time step. A current message is also published at each time step to add up the contribution of spikes to synaptic current with exponential decay.

A nodes and topics graph is generated using a ROS visualization tool. The graph is shown in Figure 3.3. We use Turtlebot 2 in Gazebo simulator to test our model. In the figure, We hide the nodes and topics related to Turtlebot 2, and use a single Gazebo node to represent all of them. At initialization state, landmark likelihood layer gives



Figure 3.3: Nodes and topics graph of HD cell model from ROS

HD cells layer a long initialize synaptic current to form a stable initial attractor state in HD cells layer. Two cue receiver layers are added in the implementation, they help Bayesian inference layer to compute synaptic current. Two visualization nodes are also implemented in the model to show the firing rate and raster map of different layers, these nodes are also used to save history informations for future analyze.

3.6 Experiments and results

Experiments for the HD cell model uses Gazebo simulator. Gazebo is a robot simulator in ROS simulates robots in customized 3D environments. Simulated robots in Gazebo simulator publish same topics and provide same services compare to real robots, this ensures the model tested on Gazebo can be transfered into a real robot without any change. We built a simulating environment shown in Figure 3.4. The environment simulates a simple room with 4 straight walls and 4 vertical corners. Similar environments have been used in the experimental studies for border cells, grid cells, and place cells [2]. In the experiments we put the Turtlebot 2 robot in the center of the environment. The robot rotates with angular velocity of 10 deg/s counter-clockwise.

Different parameters used in spiking neuron model and synaptic weights can cause huge difference in the result. Parameters in our HD cell model is shown in Table 4.1. We tune the parameters to get a well-performed LIF model. Firing rate of this LIF model increase almost linearly with the increase of current input, and very sensitive



Figure 3.4: Simulation environment in Gazebo simulator for HD cell model experiments

to current changes. Decay factor for synaptic current decides how long a spike will contribute to the output current. Too large decay factor will cause the output current very unstable, since contributions of spikes decay too fast. At the same time, too small decay factor will cause the output current unable to adapt to the change of neuron activity in time. Gaussian distributions are used for synaptic weights between layers. A global decrease of the distribution makes global inhibition for neurons far away, and variance of the distribution controls the area of excitation.

HD cell doesn't fire above baseline when the animal's head is pointing far away from HD cell's preferred firing direction, and increase its firing rate as the animal moves its head towards the preferred direction. Experimental studies show HD cell's directional firing range average around 90 degrees, and the range varies from 60 degrees to 150 degrees [39]. Tuning curves of 2 HD cells from the simulation of our HD cell model is shown in Figure 3.5. The HD cell in the simulation shows a smaller directional firing range, varies from 20 degrees to 30 degrees, compares to real experimental findings. We find smaller directional firing range helps to keep a stable attractor bump in spiking CANN model, and gives a more accurate head direction representation for further computations in mapping and localization. However, smaller range also causes problems when HD cells layer distribution and landmark likelihood distribution varies a lot, this may cause catastrophic consequences like deactivating all HD cells. LIF neuron model parameters and synaptic current parameter:

dt	au	R	Reset voltage	Spike voltage	Decay factor
10.0	60.0	1.0	-50.0	1.0	1.1

Synaptic weights for center excitatory, global inhibitory synapse connection HD cells layer and HD cells layer:

Mean	Variance	Decrease	Amplitude
0	$2\pi/(HDC \ quantity/10)$	0.4	2.0

Synaptic weights for center excitatory, global inhibitory synapse connection Bayesian inference layer and HD cells layer:

Mean	Variance	Decrease	Amplitude
0	$2\pi/(HDC \ quantity/5)$	0.5	1.0

Synaptic weights for shifted center excitatory, global inhibitory synapse connection HD transition layers and HD cells layer:

Mean	Variance	Decrease	Amplitude
± 6	$2\pi/(HDC \ quantity/10)$	0.5	1.5

Table 3.1: Parameters in simulation experiment for HD cell model

A partial raster plot for the HD cells layer neurons is shown in Figure 3.6. Spikes for HD cells with index 110 to 160 during 16.5 seconds to 18.5 seconds are shown in the plot. Red spikes are fired under the correction from Bayesian inference layer, and black spikes are fired without the correction. The plot shows less HD cells are active when correction from Bayesian inference layer is performed on HD cells layer. Since HD cells layer represents a distribution of head direction, less HD cells firing represents a more accurate distribution with smaller variance. This phenomenon in the model is cause by the correction input from the Bayesian inference layer, a Bayesian inference result with smaller variance is given to the HD cells layer every 500 milliseconds, caused the attractor bump in HD cells layers to shrink.

Accuracy of the HD cell model we developed compared to a simple CANN HD cell model is shown in Figure 3.7. A long simulation experiment (300 seconds) in Gazebo simulator has been done for both models. Previously we show, experimental studies find HD cell firing patterns drift from ground truth if no visual information is provided. A spiking CANN HD cell model without Bayesian correction shows similar



Figure 3.5: Tuning curve for 2 simulated HD cells



Figure 3.6: A partial raster map for HD cells layer neurons

drift error in our simulation experiment, the error accumulates as time goes on due to instability of spiking network and noise in the self-motion information. Compared to the spiking CANN HD cell, the HD cell model we propose eliminates the drift error using visual information. At the same time, our model also lessen the uncertainty of the visual information using self-motion information. The result shows that although visual information contributes to the accuracy of the direction representation in HD cell, they can't provide the ground truth of head direction, self-motion information are needed to rectify the errors.

We showed that a model of an HD cell system, reinforced by a neural layer that performs Bayes inference, can combine stimulus from a self-motion speed sensor and



Figure 3.7: result from a long experiment where the robot rotates with angular velocity of 10 deg/s

a visual sensor to accurately assess on the direction of the head. The addition of the visual information improved the head direction representation in the presence of noise. Although still is infancy, our robotic system mimics the behavioral abilities observed in mammals, at least in terms of localizing its head direction.

Chapter 4

Dynamic border cell mapping model

4.1 Why we built a dynamic border cell mapping model?

A navigation system relies on environmental map to perform localization and route planning. In robotics, environmental map can be simplified to positions of landmarks. Border cells have been found in the brain of most mammals. Experimental studies show border cells form a map of the environment by reacting to border-like landmarks at different preferred direction and distance [4]. Besides their special firing patterns, we still don't know much about how border cells work. There is no neurophusiological explanation on how a map is learned in a spiking border cell network using observation of the environment.

Observation of the environment from vision is always egocentric. However, though receive inputs from observation, border cells have allocentric preferred directions. There no computation model with spiking neurons explains the transform from egocentric observation to allocentric border cell activity. Border cell fires for a border-like landmark at its preferred direction and distance no matter this landmark is in the range of view or not. It means a learning process must exist inside border cell network to memorize all observed landmarks. This learning process has been neglected by most of the computational models of border cells. BVC model for border cell simply defines a known environment for the agent to generate border cells' firing patterns [19], and no spiking neurons involve in the process.

We propose a dynamic border cell mapping model composed by egocentric border cell layer and allocentric border cell layer. This model uses spiking neurons to provide environmental map to our Gridbot spiking neuron system. The transformation between egocentric layer and allocentric layer using HD cells gives an assumption of how border cells use egocentric observation to form allocentric firing patterns. The memory of environmental map inside allocentric layer is formed by synaptic plasticity on the interconnections between border cells. This is the first of its kind model for spiking border cells, and it has the potential to point the way for neuroscientists for better understanding of border cell system in the brain.

4.2 Egocentric border cells

Depth perception is a very important visual ability for human and other mobile creatures to perceive the environment in 3 dimensions. It helps animals to sense the distance of an object. Human depth perception arises from binocular vision [40], a visual system using two eyes simultaneously. This system not only gives a wider field of view, approximately 190 degrees with two eyes compares to 120 degrees with one eye, but also gives precise depth perception using the different positioned visual inputs given by two eyes.

Depth cameras used on robots are very similar to human eyes regarding to depth perception. The Asus Xtion PRO LIVE 3D sensor on Turtlebot 2 uses a stereo camera with two lens, it can sense the depth of objects in a range from 0.8 to 3.5 meters with a 60 degrees horizontal range of view. Though the depth perception from a depth cameras is a lot weaker than human eyes in terms of range of object distance and range of view, it simulates the depth perception inputs perfectly to the spiking neural network model.

Border cells in the brain can sense the border of the environment located in different distances and directions. Border cells found in experimental studies act allocentricly, their preferred directions don't follow the changes of the head direction. However, inputs from visual systems like eyes or depth cameras are egocentric, an allocentric direction of head is not needed when processing these inputs. In order to process inputs directly from depth camera and avoid using the noisy head direction representation from the HD cell model, we propose a layer of spiking egocentric border cells to encode depth information into spiking neurons.



Figure 4.1: Transform laser scan signal into spiking of neurons in egocentric border cell layer

The spiking egocentric border cell layer transforms a laser scan signal from depth camera into activities of real-time spiking LIF neurons, Figure 4.1 shows how the transformation works. Laser scan signal is a simplification from 3D point clouds generated by depth camera, it gives an estimated nearest obstacle distance for each sampled direction in the range of view. Neurons in the egocentric border cell layer have 60 different preferred head directions, ranging from -30 degrees to 30 degrees based on the egocentric head direction at 0 degree. For each direction, 5 different preferred distances ranging from 50 centimeters to 200 centimeters are defined. The preferred distance increases exponentially with $\sqrt{2}$, this makes our model generates more accurate border representation for closer borders and landmarks.

Each neuron in these 300 neurons receives a synaptic current input based on data in laser scan signal. In ROS, laser scan is presented as an array of distances for every sampled directions. Since the preferred head directions here is less than the sampled directions, we first average the laser scan array to get a averaged distance for each preferred head direction. Then we select the neuron with nearest preferred distance comparing to the averaged distance, and give it the synaptic current input respects to the difference of distance linearly. At each time step (10 milliseconds), at most 1 neuron for each preferred direction will receive inputs from vision.

4.3 Allocentric border cells

As we said before, Border cells respond to borders in the environment allocentricly. For instance, if a border cell responds to the borders in the east of the creature, it will always fire if there is a border in the east within the preferred distance, no matter what direction its head will be facing. In our model, this activity requires to transform egocentric representation of the border provided by egocentric border cell layer to an allocentric representation.

Transformation from an egocentric direction to an allocentric direction with a known head direction is not a hard job in geometry. The transformation is a circular shifting operation shifts a direction based on 0 degree to the same direction based on current head direction. However, the job appears a lot more difficult to accomplish using spiking neural networks. There is no theory about how the brain transforms between egocentric information and allocentric information, most existing models of border cells directly use the geometrical operation to get the job done [10].

We build a allocentric border cell layer with 1800 spiking border cells. Border cells in the layer have 360 preferred head directions, and 5 preferred distances. Each group of border cells with the same preferred head direction has a corresponding HD cell. Egocentric border cell layer connects with allocentric border cell layer using group synapses as shown in Figure 4.2. Synapses in one group changes simultaneously, and only one group of synapses are activated at one time. Here we propose a method to transform egocentric border information to allocentric border information in the allocentric border cell layer, the method uses HD cells layer to manipulate synaptic weights between egocentric and allocentric border cell layers.

HD cells layer in the previously proposed HD cell model gives an accurate representation of head direction. In this layer head direction is represented by a distribution formed by 360 neurons. If we assume the distribution has a similar shape to Gaussian distribution, we can directly get the mean of the distribution by locating the neuron with maximum firing rate. This neuron is used to activate the certain group of synapses between egocentric and allocentric layers.



Figure 4.2: Transform egocentric border information into allocentric border information using HD cells

$$G = \operatorname{arcmax}_{j} S_{j} \mid S_{j} = \sum_{j} w_{ij} a_{ij}, \ j = 1, ..., 360, \ i = 1, ..., 60$$

$$(4.1)$$

Equation 5.1 is used to determine which group of synapses between egocentric and allocentric layers is going to be activated, preferred distances are neglected in the equation for simplification. We define two parameters for each synapse *i* in group *j*, a_{ij} is the head direction activation and w_{ij} is the head direction weight. As shown in Figure 4.2, HD cell with maximum firing rate changes the head direction activations of the synapses connected to the corresponding border cell. Head direction weight of a synapse is precomputed by the absolute difference between post-synaptic border cell preferred direction and the center allocentric head direction of the group. The larger the difference, the smaller the head direction weight. By using equation 5.1, the group of synapses *G* with the head direction activation nearest to the center of the group will be selected. Same synaptic weights are given to all synapses in the selected group, and egocentric border cells drive allocentric border cells to spike using these activated synapses.

4.4 Mapping the environment employing synaptic plasticity

The strength of synaptic current from pre-synaptic neuron to post-synaptic neuron is determined by not only the contribution of the spikes, but also the weight of the synapse.

In previous models we proposed, weights of synapses are all predefined. Though predefined weights works perfectly in neural models, they can't adapt to unknown or changing environments. Experimental studies show synaptic weights can be strengthen (Longterm potentiation, LTP) or weaken (Long-term depression, LTD) over time by synaptic plasticity [41]. A neural model with synaptic plasticity learns weights of synapses through interaction with the environment, this can leads the model to better memorize and adapt to the environment.

$$\frac{d}{dt}w_{ij} = cv_i v_j \tag{4.2}$$

$$\frac{d}{dt}w_{ij} = c(w_{max} - w_{ij})v_iv_j \tag{4.3}$$

$$\frac{d}{dt}w_{ij} = c_1(w_{max} - w_{ij})v_iv_j - c_2w_{ij}$$
(4.4)

Hebb rule defines that neurons fire together, wire together [42]. Suppose we have two neurons, pre-synaptic neuron i and post-synaptic neuron j. At time t, neuron ihave activity v_i and neuron j have activity v_j . Equation 5.2 shows the simplest Hebbian learning rule to update the synaptic weight w_{ij} between two neurons. Weight changes in Equation 5.2 only depend on pre- and post-synaptic neurons, and the neurons have to be active simultaneously to make the changes to occur. Weight never stop increasing in the simplest rule, a soft bound w_{max} is added onto the rule in Equation 5.3 to make weight stop increasing gradually. However, Hebbian learning rule with a soft bound causes weights eventually saturate at an upper maximum value. Since there is no decrease of weight in the learning rule, this causes the neural model lost the ability of learning. Therefore, leaky term in Equation 5.4 is needed to make the learning more adaptive. The leaky term gives synapse the ability to forget learned weight and learn new weight to adapt changes in the environment.

Neuroscience studies show synaptic weights changed by long-term plasticity can last for hours or even days, this makes synaptic plasticity to be one of the important foundations of memory [41]. Memories are believed preserving in the interconnections between neurons [43]. For instance we have two neurons A and B, neuron A fires when an apple appears and neuron B fires when color red appears. When a red apple appears, the synaptic weight between A and B is strengthen over time. The memory of seeing a red apple is saved by the interconnection between A and B. Next time when an apple appears both neuron A and B will fire, and we directly know the color of the apple using memory.

Map of the environment is the memories of all landmark positions in the environment. Here in our model, memory of landmark position is represented by firing of a border cell with corresponding preferred direction and distance. Since the environment is unknown, we assume the initial position of the robot to be the coordinate origin of the map. Suppose we have 3 different border cells in allocentric border cell layer fire for 3 existing landmarks in the environment. Due to the limitation on range of view, only 2 landmarks can be observed each time. The synaptic weight between 2 border cells is strengthen when they fire together. When the robot rotates and observes the third landmark, all 3 border cells are connected together.Therefore, a cognitive map represented by border cells and their interconnections is learned.



Figure 4.3: Learning environmental map using synaptic plasticity, each neuron in allocentric border cell layer only connect with its surrounding neurons

We add synaptic plasticity inside allocentric border cell layer to memorize the environmental map. Learning process of the synaptic weights between border cells is shown in Figure 4.3. The plasticity inside the layer is designed to accomplish two goals. First goal is the persistent firing of border cells for landmarks outside the range of view. Experimental studies show border cells are activated no matter the border is in the view or not, as long as the border is in the preferred direction and distance range [4]. Second goal is the real-time spiking for all 1800 border cells inside the layer. The standard way for synaptic plasticity is to built an all connection network, each neuron has synapses connect to all other neurons. Since we need to build a real-time model, the all connection network is computational impracticable. Therefore, border cell inside the layer only has synapses with its surrounding neurons, this makes synaptic plasticity plausible for real-time processing.

$$\tau \frac{d}{dt} v^{allo} = -v + R(w_{mn}^{ego} I_{mn}^{ego} + \sum_{surround} w_s^{allo} I_s^{allo})$$
(4.5)

Updating rule for border cell in allocentric border cell layer is shown in Equation 5.5. Each border cell receives synaptic current inputs from two sources, the corresponding egocentric border cell with visual information and the surrounding border cells with memory information. When the landmark is inside the range of view, current inputs from egocentric border cell dominates other inputs. At the same time, synaptic weight w_m^{allo} increases if both border cell and its surrounding border cell s are activated.

When the robot rotates and the landmark moves outside the range of view, border cell only receives current inputs from its surrounding border cells. Synapses between border cells are symmetrical. A spiking border cell gives synaptic current to its surrounding border cells, and contributes to their spiking. Surrounding spiking border cells also give synaptic currents to this border cell to maintain its spiking activity. This symmetrical behavior inside the allocentric border cell layer preserves the position of observed landmarks in small groups of spiking border cells without any external visual information inputs.

4.5 Model implementation in ROS

Same with previous HD cell model, we implement our border cell model in ROS. A nodes and topics graph shown in Figure 4.4 is generated us ing a ROS visualization tool. In addition to the nodes and topics in Border Cell model, the graph also includes the nodes and topics from HD Cell model to show the interaction between two models. At initialization state, allocentric border cell layer uses 0 degree head direction and waits for messages from HD cells layer. Two visualization nodes are implemented in the model to show the firing activity of egocentric border cells and allocentric border cells.



Figure 4.4: Nodes and topics graph of border cell model from ROS

4.6 Simulation and results

Experiments for the Border Cell model uses both Gazebo simulator and real Turtlebot 2 robot. In both simulator and real world environment, robot rotates with angular velocity of 10 deg/s counter-clockwise. Robot is banned to move its location during the experiment before the map is learned, because the map learned by the model is based on the initial location of the robot. For the experiment on real Turtlebot 2

Parameters for Egocentric Border Cell Layer:

Prefer distance	Prefer direction	Max current	Current decay slope
5	60	20.0	20.0

Parameters for Allocentric Border Cell Layer:

Prefer distance	Prefer direction	Ego to Allo weight
5	360	10.0

Parameters for synaptic plasticity in Allocentric Border Cell Layer:

dt	Max synaptic weight	c_1	c_2
10.0	5.0	0.01	0.0005

Table 4.1: Parameters in simulation experiment for Border Cell model

robot, we setup a local area network connecting the ROS system on the robot and our Desktop computer. This allows us to remotely control the robot using a more powerful computer.



Figure 4.5: Experiment environments for border cell model

We use the same LIF neuron in both Border Cell model and HD Cell model. All other parameters assigned in Border Cell model is shown in Table 5.1. Synaptic current input to egocentric border cell layer decrease from maximum current base on difference between preferred and actual distance of the landmark. The decrease of input current follows a linear decay defined by the current decay slope. The selected group of synapses from egocentric to allocentric border cell layer are given the same synaptic weight. Synaptic plasticity in allocentric border cell layers using Equation 5.4 to update the surrounding synaptic weights between border cells, Euler's method is used here to solve the differential equation.

Except the simulation environment we use for HD cell model experiments previously,

we built 2 other environments for Border Cell model experiments to give a comprehensive test on the model. The environments are shown in Figure 4.5. On the left side is a new environment in Gazebo simulator with two separate objects. This environment aims to test mapping on not connected landmarks, different with the environment we use for HD cell model experiments. On the right side is a real world environment we built in our lab. By using this environment, we want to test if our model works correctly on real robots.



Figure 4.6: Experimental map generated in 3 different environments, colors in the map represents strength of border cell activity, x-axis represents preferred directions and y-axis represents preferred distances. a) Simulator environment of room. b) Simulator environment of separate objects. c) Real world environment in the lab.

Environment maps generated by the Border Cell model for 3 different environments are shown in Figure 4.6. The strength of border cell firing activities is represented by colors in the maps. The figure on the top is the border cell map for the simple room environment. The corners of the room are captured by the spiking of border cells with different preferred distances. The figure in the middle is the border cell map for the separate objects environment. The map proves the model can memorize an observed landmark even if no part of the landmark is in the range of view. The last figure is the border cell map for the real world environment. Real world environment is more complicated than environments built in Gazebo simulator. The map shows both the Border Cell model and the HD Cell model work perfectly on real robots.

We showed that a model of border cell mapping system can use spiking of neurons to memorize and encode a map the environment. The model works perfectly in both simulator environments and real world environment. Although the neurologic theory behind how border cells generate their firing patterns is still unknown, our model gives a feasible hypothesis by mimicking the behaviors observed in mammals on border detection and environment sensing.

Chapter 5

Conclusion

In this thesis we proposed a biologically feasible model for HD cell and border cell. For HD cell model, we developed the first model for HD cell using spike based Bayesian inference. For border cell model, we gave an assumption of how environmental map is formed inside border cell network by using inputs from depth camera. Our model gain good results in the simulation, and is proved working in real world environment.

However, there are also some limitations in our current model for mapping the environment. Our model propose in this thesis doesn't consider movements of robot. Robot can only build map for one position by rotating 360 degrees. If robot changes its position, the environmental map built before won't follow the change. This is because the model now lack a transition mechanism for the map. A transition system will be developed in the future to make the map in border cell network to move itself whenever robot changes its position.

Another limitation in the model is the artificial input used by HD cell landmark likelihood layer. This artificial input can be replaced by many kinds of real visual inputs. The best replacement for this input is the border cell activity. The future plan for solving this limitation is to use both egocentric and allocentric border cells to compute a spiking based likelihood distribution.

Though still having these limitations, our model proposed here makes the foundation of Gridbot. Future steps for building the Gridbot will focus on two parts. First we will generalize the border cell model and the HD cell model to make direction information in HD cells and environmental map in border cells update simultaneously. By connecting to each other, a noise filtering system can be encoded into a biologically feasible spiking neural network. Second we will extend current 1D model into 2D environment by proposing models for grid cell and place cell. Multiple grid cells with different grid spacing can be used together to represent locations. And connected place cells can be used to represent trajectories. Eventually Gridbot system will be built more flexible to adapt to new findings in neuroscience.

As an overarching goal to fully mimicking the brain spatial system, Gridbot system is an exciting approach we will be working on in the near future. We hope Gridbot system to point the way for neuroscientists to uncover the brain spatial system, and to bring new ideas and methods to robotic field.

Bibliography

- MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241, 2001.
- [2] John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. Brain research, 34(1):171–175, 1971.
- [3] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.
- [4] Trygve Solstad, Charlotte N Boccara, Emilio Kropff, May-Britt Moser, and Edvard I Moser. Representation of geometric borders in the entorhinal cortex. *Science*, 322(5909):1865–1868, 2008.
- [5] JB Ranck Jr. Head direction cells in the deep cell layer of dorsal presubiculum in freely moving rats. In Soc Neurosci Abstr, volume 10, 1984.
- [6] Emilio Kropff, James E Carmichael, May-Britt Moser, and Edvard I Moser. Speed cells in the medial entorhinal cortex. *Nature*, 523(7561):419–424, 2015.
- [7] Edmund T Rolls et al. Spatial view cells and the representation of place in the primate hippocampus. *Hippocampus*, 9(4):467–480, 1999.
- [8] SM Stringer, ET Rolls, TP Trappenberg, and IET De Araujo. Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells. *Network: Computation in Neural Systems*, 13(4):429–446, 2002.

- [9] Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network models of grid cells. PLoS Comput Biol, 5(2):e1000291, 2009.
- [10] Allen Cheung. Probabilistic learning by rodent grid cells. *PLoS Comput Biol*, 12(10):e1005165, 2016.
- [11] Kechen Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, 16(6):2112– 2126, 1996.
- [12] Albert Rose. The sensitivity performance of the human eye on an absolute scale. JOSA, 38(2):196–208, 1948.
- [13] Dora E Angelaki and Kathleen E Cullen. Vestibular system: the many facets of a multimodal sense. Annu. Rev. Neurosci., 31:125–150, 2008.
- [14] Wei Ji Ma, Jeffrey M Beck, Peter E Latham, and Alexandre Pouget. Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432– 1438, 2006.
- [15] Mark F Bear and Robert C Malenka. Synaptic plasticity: Ltp and ltd. Current opinion in neurobiology, 4(3):389–399, 1994.
- [16] Hanne Stensola, Tor Stensola, Trygve Solstad, Kristian Frøland, May-Britt Moser, and Edvard I Moser. The entorhinal grid map is discretized. *Nature*, 492(7427):72– 78, 2012.
- [17] Linda Palmer and Gary Lynch. A kantian view of space. Science, 328(5985):1487– 1488, 2010.
- [18] John Widloski and Ila R Fiete. A model of grid cell development through spatial exploration and spike time-dependent plasticity. *Neuron*, 83(2):481–495, 2014.
- [19] Kate Jeffery and Neil Burgess12. The boundary vector cell model of place cell firing and spatial memory. *Reviews in the Neurosciences*, 17:71–97, 2006.

- [20] Neil Burgess, Caswell Barry, and John O'keefe. An oscillatory interference model of grid cell firing. *Hippocampus*, 17(9):801–812, 2007.
- [21] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*, pages 593–598, 2002.
- [22] Thomas Trappenberg. Continuous attractor neural networks. Recent developments in biologically inspired computing, pages 398–425, 2003.
- [23] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 79(8):2554–2558, 1982.
- [24] A David Redish, Adam N Elga, and David S Touretzky. A coupled attractor model of the rodent head direction system. Network: Computation in Neural Systems, 7(4):671–685, 1996.
- [25] Alexis Guanella, Daniel Kiper, and Paul Verschure. A model of grid cells based on a twisted torus topology. *International journal of neural systems*, 17(04):231–240, 2007.
- [26] Theocharis Kyriacou. An implementation of a biologically inspired model of head direction cells on a robot. In *Conference Towards Autonomous Robotic Systems*, pages 66–77. Springer, 2011.
- [27] Thomas Degris, Loic Lacheze, Christian Boucheny, and Angelo Arleo. A spiking neuron model of head-direction cells for robot orientation. In Proceedings of the eighth int. conf. on the simulation of adaptive behavior, from animals to animats, pages 255–263, 2004.
- [28] John O Keefe and Neil Burgess. Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425, 1996.

- [29] Jeremy P Goodridge, Paul A Dudchenko, Kimberly A Worboys, Edward J Golob, and Jeffrey S Taube. Cue control and head direction cells. *Behavioral neuroscience*, 112(4):749, 1998.
- [30] Jeffrey S Taube. Head direction cells and the neurophysiological basis for a sense of direction. *Progress in neurobiology*, 55(3):225–256, 1998.
- [31] RHS Carpenter and MLL Williams. Neural computation of log likelihood in control of saccadic eye movements. *Nature*, 377(6544):59, 1995.
- [32] Martin Boerlin and Sophie Deneve. Spike-based population coding and working memory. PLoS Comput Biol, 7(2):e1001080, 2011.
- [33] Rajesh PN Rao. Hierarchical bayesian inference in networks of spiking neurons. In NIPS, pages 1113–1120, 2004.
- [34] Eugene M Izhikevich. Which model to use for cortical spiking neurons? IEEE transactions on neural networks, 15(5):1063–1070, 2004.
- [35] Rajesh PN Rao. Bayesian computation in recurrent neural circuits. Neural computation, 16(1):1–38, 2004.
- [36] Ning-long Xu, Mark T Harnett, Stephen R Williams, Daniel Huber, Daniel H OConnor, Karel Svoboda, and Jeffrey C Magee. Nonlinear dendritic integration of sensory and motor input during an active sensing task. *Nature*, 492(7428):247–251, 2012.
- [37] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, 2009.
- [38] Kathryn Eleda Brenan, Stephen L Campbell, and Linda Ruth Petzold. Numerical solution of initial-value problems in differential-algebraic equations. SIAM, 1995.
- [39] Jeffrey S Taube. The head direction signal: origins and sensory-motor integration. Annu. Rev. Neurosci., 30:181–207, 2007.

- [40] Rebecca K Jones and David N Lee. Why two eyes are better than one: the two views of binocular vision. Journal of Experimental Psychology: Human Perception and Performance, 7(1):30, 1981.
- [41] SF Cooke and TVP Bliss. Plasticity in the human central nervous system. Brain, 129(7):1659–1673, 2006.
- [42] Carla J Schatz. The developing brain. Scientific American, 267(3):60–67, 1992.
- [43] Donald Olding Hebb. The organization of behavior: A neuropsychological theory. Psychology Press, 2005.