EXPLORATION OF NEURAL CODING
IN RAT’S AGRANULAR MEDIAL AND AGRANULAR LATERAL CORTICES
DURING LEARNING OF A DIRECTIONAL CHOICE TASK

by

Hongwei Mao

A Dissertation Presented in Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy

Graduate Supervisory Committee:

Jennie Si, Chair
Christopher Buneo
Yu Cao
Marco Santello

ARIZONA STATE UNIVERSITY

May 2014
ABSTRACT

Animals learn to choose a proper action among alternatives according to the circumstance. Through trial-and-error, animals improve their odds by making correct association between their behavioral choices and external stimuli. While there has been an extensive literature on the theory of learning, it is still unclear how individual neurons and a neural network adapt as learning progresses.

In this dissertation, single units in the medial and lateral agranular (AGm and AGl) cortices were recorded as rats learned a directional choice task. The task required the rat to make a left/right side lever press if a light cue appeared on the left/right side of the interface panel.

Behavior analysis showed that rat’s movement parameters during performance of directional choices became stereotyped very quickly (2-3 days) while learning to solve the directional choice problem took weeks to occur. The entire learning process was further broken down to 3 stages, each having similar number of recording sessions (days). Single unit based firing rate analysis revealed that 1) directional rate modulation was observed in both cortices; 2) the averaged mean rate between left and right trials in the neural ensemble each day did not change significantly among the three learning stages; 3) the rate difference between left and right trials of the ensemble did not change significantly either. Besides, for either left or right trials, the trial-to-trial firing variability of single neurons did not change significantly over the three stages. To explore the spatiotemporal neural pattern of the recorded ensemble, support vector machines (SVMs) were constructed each day to decode the direction of choice in single trials. Improved
classification accuracy indicated enhanced discriminability between neural patterns of left and right choices as learning progressed. When using a restricted Boltzmann machine (RBM) model to extract features from neural activity patterns, results further supported the idea that neural firing patterns adapted during the three learning stages to facilitate the neural codes of directional choices.

Put together, these findings suggest a spatiotemporal neural coding scheme in a rat AGl and AGm neural ensemble that may be responsible for and contributing to learning the directional choice task.
ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Dr. Jennie Si for being a great teacher, mentor and leader. Her guidance opened up the world of neuroscience to me and her support led me all the way through the work of my research and this dissertation. Her enthusiasm, diligence and persistence will continue encouraging me.

I want to thank Dr. Glen P. Abousleman for providing exciting project opportunities. Throughout those projects, I received valuable training on programming and software engineering skills, problem solving and writing.

Thanks to my dissertation committee members, Dr. Marco Santello, Dr. Christopher Buneo and Dr. Yu Cao, for their constructive advices.

I also want to thank former lab member Changchun Li and Chenhui Yang for their guidance when I first joined the lab. Thanks to Yuan Yuan and other lab members for their assistance and cooperation on the research. Additionally, without the financial support from the National Science Foundation and General Dynamics C4 Systems this work would not have been possible.

Last but not least, I thank my parents, my sister and the rest of my family for their support through my educational journey.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Learning and neural adaptation</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Related neural network</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 2 THE DIRECTIONAL CHOICE TASK</td>
<td>7</td>
</tr>
<tr>
<td>2.1 The behavioral task</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Extracellular recording</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Behavioral results</td>
<td>11</td>
</tr>
<tr>
<td>2.4 Neural modulations by choice direction</td>
<td>15</td>
</tr>
<tr>
<td>2.5 Conclusions</td>
<td>21</td>
</tr>
<tr>
<td>CHAPTER 3 SPATIOTEMPORAL FIRING PATTERN</td>
<td>23</td>
</tr>
<tr>
<td>3.1 Support vector machine decoder</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Neural pattern modeling using SVMs</td>
<td>26</td>
</tr>
</tbody>
</table>
3.3 Firing pattern adaptation with learning .................................................. 29
3.4 Discussions ................................................................................................. 34

CHAPTER 4 FIRING VARIABILITY ANALYSIS .................................................... 41
4.1 Trial-to-trial firing variability during learning ............................................. 41
4.2 Dynamics of firing variability according to task demands ......................... 43
4.3 Firing variability of two distinct task factors .............................................. 45
4.4 Trial-to-trial firing rate fluctuation .............................................................. 49
4.5 Summary ....................................................................................................... 50

CHAPTER 5 MODELING NEURAL FIRING PATTERNS THROUGHOUT THE
LEARNING PROCESS ......................................................................................... 52
5.1 Energy-based models and restricted Boltzmann machines ....................... 52
5.2 Neural firing pattern modeling using RBMs .............................................. 58
5.3 Firing pattern adaptation with learning ..................................................... 63
5.4 Conclusion ................................................................................................... 69

CHAPTER 6 CONCLUSION ............................................................................... 71
REFERENCES .................................................................................................... 73
APPENDIX ........................................................................................................ 81
A AUTOMATIC MOVING TARGET TRACKING ............................................ 81
A.1 Introduction ................................................................................................. 83
A.2 System Overview and Foreground Detection ............................................ 86
A.3 Data association ................................................................. 93
A.4 Track management ............................................................ 100
A.5 Experimental results ........................................................ 105
A.6 Conclusions .................................................................. 111
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The behavioral task, trial timeline, and recording sites.</td>
</tr>
<tr>
<td>2.</td>
<td>Example single unit recordings in three sessions from rat A09.</td>
</tr>
<tr>
<td>4.</td>
<td>The three learning stages.</td>
</tr>
<tr>
<td>5.</td>
<td>Response latency to start a trial.</td>
</tr>
<tr>
<td>6.</td>
<td>Raster and peri-event histograms of example neurons.</td>
</tr>
<tr>
<td>7.</td>
<td>Dynamics of firing rate differences between L-L and R-R trials.</td>
</tr>
<tr>
<td>10.</td>
<td>Classification performance improved over sessions.</td>
</tr>
<tr>
<td>11.</td>
<td>SVM classification performance and prediction accuracy.</td>
</tr>
<tr>
<td>13.</td>
<td>Dynamics of firing rate variability in the period of one trial.</td>
</tr>
<tr>
<td>14.</td>
<td>Both task factors were coded in motor cortical neurons.</td>
</tr>
<tr>
<td>15.</td>
<td>Direction selective neurons exhibited smaller variability than previous outcome selective neurons.</td>
</tr>
<tr>
<td>16.</td>
<td>Trial-to-trial firing rate fluctuation explained part of the variation.</td>
</tr>
<tr>
<td>17.</td>
<td>Illustration of a restricted Boltzmann machine (RBM) and a deep belief network (DBN).</td>
</tr>
<tr>
<td>18.</td>
<td>Example learning curves of RBM training.</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>19. DBNs extracted features beneficial to direction information decoding.</td>
<td>61</td>
</tr>
<tr>
<td>20. Classification performance under different RBM size.</td>
<td>62</td>
</tr>
<tr>
<td>21. Classification performance with versus without RBM processing.</td>
<td>63</td>
</tr>
<tr>
<td>22. One model to decode individual session’s data.</td>
<td>64</td>
</tr>
<tr>
<td>23. Classification performance using single-session models versus one model built on all sessions’ data.</td>
<td>66</td>
</tr>
<tr>
<td>A1. A typical multiple target tracking system.</td>
<td>87</td>
</tr>
<tr>
<td>A2. Foreground detection.</td>
<td>91</td>
</tr>
<tr>
<td>A3. Foreground detection under target interactions.</td>
<td>92</td>
</tr>
<tr>
<td>A4. An illustration of overlap-rate-based data association.</td>
<td>95</td>
</tr>
<tr>
<td>A5. Overview of track updating.</td>
<td>103</td>
</tr>
<tr>
<td>A6. Targets undergo substantial appearance changes.</td>
<td>108</td>
</tr>
<tr>
<td>A7. Tracking through complete occlusion.</td>
<td>108</td>
</tr>
<tr>
<td>A8. Target interactions.</td>
<td>110</td>
</tr>
<tr>
<td>A9. Tracked target trajectories.</td>
<td>111</td>
</tr>
</tbody>
</table>
CHAPTER 1 INTRODUCTION

1.1 Introduction

In real world, people often face the situation of choosing a proper action among alternatives according to the circumstance. People make such choices usually depending on the consequences of choosing different options which they have learned from previous experience. Such learning process involves executive function skills such as decision making and error correction which are essential functions of the brain. While there has been an extensive literature and several schools of thoughts on the theory of learning, it is still unclear how individual neurons and a neural network adapt as learning progresses.

Using an experimental task setup, such learning process could be conducted with animals through operant conditioning. Animals perform a task in which different sensory stimuli are presented one at a time. Accordingly, they choose to execute one of the potential actions, but only certain action in response to a particular stimulus would be rewarded. Under this reinforcement learning paradigm, the odds for animals to perform the rewarded stimulus-response pairs would increase over time.

The behavior of choosing the rewarded action in response to a stimulus could happen throughout the learning process. Although such behavior could be externally identical, the same sensory input followed by the same action, the internal mental state could be quite different. Early in the learning process when a subject is naïve, correctly choosing the rewarded action happens by chance possibly under the strategy of trial and error. When the multi-choice task has been mastered, action selection could be made by applying the learned stimulus-response associations. Such differences in the way that a
choice is made emerge with learning, but the neural correlate that underlies this learning process is not fully understood. In this dissertation, neural adaptation during learning of a two-choice task was studied using a rat model to explore the neural code as learning progresses.

1.2 Learning and neural adaptation

The brain serves as the center of the nervous system and is the most complex organ in a human body. A typical human brain contains tens of billions of neurons, and neurons are connected to one another through a structure named synapse. Signals are passed through synapses by which means neurons communicate with each other. Synaptic plasticity, the ability to strengthen or weaken a synapse, is widely assumed to be the mechanism underlying learning and memory. And memory is encoded and stored by changing the strength of connection between neurons during the course of learning. Long-term potentiation (LTP) is one of several phenomena of synaptic plasticity, and spike-timing-dependent plasticity (STDP) is a process that potentially explains LTP. While studies support the notion that synaptic plasticity is necessary for learning and memory, little is known regarding its sufficiency (Martin et al., 2000).

Besides modification of strength of existing synapses, synaptic reorganization has also been observed to associate with learning and memory. Learning novel motor tasks promotes dendritic spine formation in the motor cortex (Xu et al., 2009; Yang et al., 2009) and increases the number of synapses per neuron (Kleim et al., 2004). These changes
could further support a functional reorganization of the motor cortex through motor learning (Kleim et al., 1998, 2004).

At the neuronal level, single neurons fires action potentials (or spikes), rapid electrical pulses, to communicate with each other. And characteristics of neuron’s firing activity could be shaped by learning. In some early studies (Mitz et al., 1991; Chen and Wise, 1995), increased discharge rates during the course of learning were observed in monkey’s premotor (PM) neurons. Learning visuomotor mapping changed neuronal firing rate profile in the primary motor and non-primary motor areas (Wise et al., 1998; Paz et al., 2003), and so did motor skill learning when adapting to novel external forces (Li et al., 2001; Rokni et al., 2007).

Neural firing activity is often modeled as a stochastic process and sampled using repeated trials. The trial-to-trial firing variability could also carry task related information and might change during learning. A decline in trial-to-trial firing variability was observed after stimulus onset under a wide range of scenarios (Churchland et al., 2010). This variability could be related to other task related factors such as motor preparation and could be an indicator of neuron’s engagement in task demands (Churchland et al., 2006; Hussar and Pasternak, 2010). When monitored during a learning process, variability decreased as a motor skill improved (Kargo and Nitz, 2004).

In addition to these first and second order statistics widely used to describe neuronal firing characteristics, more sophisticated representations of firing activity have been adopted especially to decode information from a group of neurons. Laubach et al. (2000) used fine temporal patterns of neural ensembles to predict the outcome (correct
vesus wrong) of single trials. In a recent study, Huber et al. (2012) decoded licking rate of mice using neural population signal observed in several frames of imaged calcium activity. In these cases, the temporal dynamics of neural activity from a population of spatially distributed neurons formed a spatiotemporal representation of brain activity. And in both cases, the accuracy of decoding task related parameters using such spatiotemporal neural representations increased as learning progressed.

For signals as sophisticated as brain activity, many more tools have been developed to examine neural signals at multiple scales both temporally and spatially. Spike train modeling methods explore the influence of previous firing history on a neuron’s instant firing probability at a very precise temporal resolution (Okatan et al., 2004; Truccolo et al., 2005, 2010). Correlated activity measures coupling among neurons (Komiyama et al., 2010). Brain imaging techniques like functional magnetic resonance imaging (fMRI) monitors neural activity with a coarse spatial and temporal resolution but covers a large area. All these methods, among others, could provide insights regarding differential involvements of neurons, neural ensembles and brain regions during certain learning process, and benefit our understanding of the biological neural network.

Within this spectrum of neural measurements, from the cellular level to the scope of networks, monitoring extracellular single-unit firing activity offers good spatiotemporal resolution: The events of firing an action potential are monitored with a millisecond resolution, and the current multiple-electrode technology facilitates the growth in the number of simultaneously recorded neurons (Stevenson and Kording, 2011). In this work, firing activity of a population of neurons was recorded and analyzed, trying to provide an
additional data point on how individual neurons and a neural network adapt as learning progresses.

1.3 Related neural network

Associative learning has been ubiquitously evidenced in a wide range of taxa, and has been argued as a potential mechanism to support behavioral adaptation (Heyes, 2012). Learning sensorimotor mapping, in specific, associates sensory inputs with appropriate actions, and its neural correlates have been studied in not only sensory (Wiest et al., 2010; Jeanne et al., 2013) and motor (Cohen and Nicolelis, 2004; Komiyama et al., 2010; Huber et al., 2012) cortices, but also other frontal (Mitz et al., 1991; Asaad et al., 1998) and subcortical areas (Jog et al., 1999; Brasted and Wise, 2004; Pasupathy and Miller, 2005) in nonhuman primates, rodents, and other species.

A neural network that underlies visuomotor mapping has been largely identified previously (Murray et al., 2000), including the premotor (PM) cortex, the prefrontal (PF) cortex, the hippocampal system, and the basal ganglia. During learning of visuomotor associations, although striatum activity quickly reflected forthcoming motor responses, prefrontal changes were slower but stronger correlated with behavioral performance improvements (Pasupathy and Miller, 2005). Within the frontal lobes, a rostrocaudal hierarchical organization in cognitive control has been hypothesized (for a review, Badre, 2008). In particular, the premotor cortex holds stimulus-response representations and also exhibits effect of contextual information through the top-down control from prefrontal regions (Koechlin et al., 2003; Fluet et al., 2010). Human fMRI further showed that
ventral premotor area was involved in learning associations and dorsal premotor participated in execution of learned rules (Boettiger and D’Esposito, 2005). Besides, the fact that stimulation of premotor cortex enhanced mirror motor facilitation also suggested a role of PM in mirror motor effects potentially via visuomotor associative learning (Catmur et al., 2011). On the other hand, the primary motor cortex has been found to code information beyond movement kinematics (Carpenter et al., 1999; Matsuzaka et al., 2007), including features of visual stimuli which are behaviorally relevant (Zach et al., 2008; Eisenberg et al., 2011). Thus, accumulating evidence suggests the premotor and primary motor cortices as important regions to study the neural substrate underlying sensorimotor associative learning.

In the rat brain, the existence of a counterpart of primate’s prefrontal cortex is still debated, with some proposed the medial agranular area (AGm) in rat has some dorsolateral prefrontal like features (Uylings et al., 2003; Dalley et al., 2004). However, AGm’s dense and direct projection to the spinal cord makes this idea unlikely. The AGm area was proposed to be possibly homologous to the premotor cortex, supplementary motor area and frontal and supplementary eye field in primates (Reep et al. 1987; Passingham et al., 1988; Reep et al. 1990). And the lateral agranular area (AGl) has been considered homologous to the primary motor cortex (Donoghue and Wise 1982; Donoghue and Parham 1983). So in the rat brain, it points to AGm and AGl areas as candidates to observe sensorimotor mapping related activity. Previous lesion studies support this view (Passingham et al., 1988; Winocur and Eskes, 1998), and neurophysiological methods were used here to further the investigation.
CHAPTER 2 THE DIRECTIONAL CHOICE TASK

A directional choice task was developed for rats to learning certain stimulus-response associations. All procedures are in accordance with guidelines of the National Institutes of Health and approved by the institutional Animal Care and Use Committee at Arizona State University.

2.1 The behavioral task

Animals were freely moving in the chamber, which has a front panel as shown in Figure 1A. They were self-paced to start a trial by pressing the center ready lever. The center lever would retract as soon as it was pressed, and simultaneously one of the five cue lights (from left to right, LL, L, C, R and RR) would be on. The left and right control levers would extend together at 2 seconds after cue light onset and wait for animal’s response. Pressing the left control lever once would “move” the light one position to the right and pressing the right control lever would “move” the light to the left. Multiple responses to one or both levers were allowed within each single trial. The response period ended when it had been 1 second that no response was received, or when the light “moved” out of boundary (e.g. pressing the right lever when the left most light LL was on). Then a decision about trial outcome was made: a success if the center light C was on and a failure otherwise. Simultaneously, a feedback tone was played, a low tone (1 kHz) in case of a success and a high tone (12 kHz) if not. A sugar pellet reward was delivered 0.5 second after feedback tone in each successful trial. The inter-trial interval was 10 seconds for successful trials and 15 seconds for failed trials. The five cues were presented
in a pseudo-randomized fashion, and initialized with equal chances of presence at the beginning of each session. Their chances were adjusted every 25 trials and biased to generate more trials of cues with relatively lower behavioral accuracies.

Figure 1. The behavioral task, trial timeline, and recording sites. A, The rat was cued by an LED light in one of five possible positions (LL, L, C, R, and RR) in any given trial, and could use the left and right control lever to “move” light to the right and left by one position, respectively. The goal was to turn on the center light in order to receive a sugar pellet reward. B, The rat would start any trial by pressing the center ready lever at his own will, and simultaneously the light cue would present. Then there would be a 2 seconds cue-on period and after that the two control levers would extend together. The rat would respond to control levers and light position would change accordingly. A feedback tone would be played at trial end indicating the outcome. C, A 2x8 microwire array was chronically implanted in the left hemisphere of the brain, aiming for layer V pyramidal neurons in the AGm and AGl cortices.
2.2 Extracellular recording

Rats (male, Long-Evans) arrived at the age of about two weeks (with weights of around 50 grams) and were handled daily by experimenters to get accustomed to humans. They started working on a pre-training task after reached the age of 3 months and a weight of above 200 grams. The pre-training apparatus is similar to that used for recording to help animals get acclimatized to the recording chamber environment and learn the motor skills of lever pressing. After achieved a behavioral accuracy of 90% or above for at least 3 consecutive days on the pre-training task and gained weight to over 400 grams, animals were ready to receive the electrode implantation surgery.

Before surgery, rats were anesthetized by an intramuscular injection of KXA (10 mg/ml ketamine, 2 mg/ml xylazine, and 0.1 mg/ml acepromazine; 0.1cc/100g), shaved in the incision area, and placed in a stereotaxic frame. A heating water blanket was used to maintain body temperature at around 35°C, and heart rate and oxygen level were monitored through the surgery with a pulse oximeter.

Craniotomy was performed in the motor cortical areas of the left hemisphere of the brain. Microarray was centered at 2 mm lateral and 3 mm rostral from the bregma, and lowered about 1.8-2.3 mm in depth underneath dura, aiming for layer V pyramidal neurons. A headcap was formed using acrylic surrounding the electrodes and was fixed to the skull with three screws. A subcutaneous injection of 0.1 ml meloxicam was given for pain relief after surgery, and follow-up meloxicam shoots were given for the following 2 consecutive days. The rats had 7-10 days or longer as needed to recover before they were food restricted for recording sessions.
After recovered from surgery, a session of about 50 minutes was recorded daily per rat while he performed the directional control task. Animals were food restricted to get motivated to work on the task. Their weights were closely monitored to be kept up to the target weights at their ages.

The implanted microwires were arranged in a $2 \times 8$ matrix, with 500 μm or 375 μm row separation, and 500 μm electrode spacing. The polyimide-isolated tungsten microwires were 50 μm in diameter and 5 mm in length, with tips cut at 60 degrees (TDT Inc., FL). 16 channels of raw waveforms were recorded simultaneously using a RX5 Pentusa Base Station or a RX7 Microstimulator Base Station (TDT Inc., FL). Neural signals picked up by electrodes were passed to a unity gain preamplifier (bandpass 2.2 Hz ~ 7.5 kHz) through an Omnetics or a ZIF-Clip headstage, and then sampled and stored at 24.414 kHz by the base station.

Action potentials were detected and sorted off-line using the M-Sorter software (Yuan et al., 2012) based on the multiscale correlation of wavelet coefficients (MCWC) detection algorithm (Yang et al., 2011) and template matching classification. One isolated units was extracted from each of the electrodes which clearly picked up spikes (Figure 2). And the spike wave forms of all sorted neurons were inspected by experimenters to confirm their quality. According to the sites of implanted electrodes, recorded single units located in the AGl and AGm areas of the rat brain (Paxinos and Watson, 2005), and involved forelimb, neck and vibrissae representation areas (Neafsey et al., 1986; Remple et al., 2001). Microstimulations largely confirmed the sites: Whisker movements were
evoked by the minimum intensity of electrical stimulation (60 μA) through electrodes in AGl area but not those in AGm area.

Figure 2. Example single unit recordings through the same electrode in three sessions from rat A09. Top row: spike waveforms and their mean (white curve). Bottom row: inter-spike interval (ISI) distributions

2.3 Behavioral results

Male Long-Evans rats (n=8) started learning the directional control task by trial and error from a naïve state. Behavioral accuracy in each recording session was monitored and calculated as the number of correct trials over the total number of trials in that session. Rats gradually improved the accuracy over sessions, from 23.96% (average, range from
12.20% to 35.71%) on session 1 to 73.11% (average, range from 55.07% to 91.41%) on session 25 (Mann-Whitney U test, p<0.001; Figure 3A). It took 21 sessions on average for the rats to reach an accuracy of 70% or higher. Among the eight rats, two of them (A09 and I10) barely reached 70% accuracy within recorded sessions, all other six rats got to above 80%, and two rats (W09 and O10) even achieved over 90% accuracy.

Figure 3. Behavioral results. A, Rats (n=8) improved behavioral accuracy to a level above 70% in 21 days on average. B, The response latency to control levers decreased mainly during the first 3 days and stayed relatively stable afterwards.

Each animal’s learning curve was further divided into three learning stages, with approximately equal number of sessions in each stage (Figure 4). Through the entire learning process, a rapid learning occurred in learning stage I, while improvements of behavioral accuracy were slower in the latter two stages.

According to the time scheme of the directional control task, the rat had a 2 second period of time after cue onset to reach to a control lever and get prepared if he
decided to press it. He had another 1 second time to press the control lever before time was up. The response latency to control levers was calculated as the time delay from control lever extension to the first press on either lever. This latency decreased from 0.45±0.20 s in session 1 to 0.29±0.14 s in session 8 (mean±std; ANOVA, p<0.00001; Figure 3B). It got stabilized within the early stage of learning mainly during the first 3 days, and its mean was not significantly different in the following sessions compared with day 8 (ANOVA, p>0.05; one exception, day 14, p = 0.046).

Figure 4. The three learning stages. Each circle represents one session from the six rats reached 80% accuracy. The blue curve is the 5th order polynomial fit of the learning curve. Session numbers are normalized for each animal. The two dashed vertical lines indicate the division between learning stages.

It was hypothesized that over training would change a behavior from being goal directed to a habit (Daw and Shohamy, 2008). In this experiment, correctly performed
trials were rewarded with sugar pellets, so animals learned the task with the goal of earning food. Moreover, according to the task design, animals would start any trial at their own will. They would press the center ready lever to start a trial and there was no constrain of how soon they should press it after its extension. Although we did not use the standard reward devaluation test to change animal’s motivation for food, the ready lever response latency in this experiment provided a similar measure. We reasoned that the latency from ready lever extension to being pressed would be shorter when animals were hungry at the beginning of a session due to food restriction, and would be longer towards the end of a session when animals have been fed to some level of satiety. To test this, the response latency to the ready lever was analyzed in the third learning stage. Only those latencies of trials following a correct trial were used to rule out any impact from the outcome of the previous trial. These trials were divided into 25-trial blocks from the beginning to the end of a session. Figure 5 showed the statistics of this latency using rat O10 as an example. The median increased significantly from 0.50 s in block 1 to 1.08 s in block 6 (ranksum test, p<0.001). Similar trends were observed in other animals as well. This extended response latency at the end of a session indicates a declined motivation to perform the task likely due to satiety of food. Given that these results were obtained from the last learning stage, it is unlikely that animal’s behavior became habitual instead of goal-directed by the end of the learning process studied in this experiment.

The animal’s behavioral activity in the task apparatus was monitored and recorded using an infrared camera. Though free to move, rats performed stereotypical movements when they became familiar with the task. A post hoc analysis of animal’s movement trajectory confirmed this observation. Directional movements typically started
from around 300 ms after cue onset, and the duration lasted for about 700 ms. In the next
around 1 second time period, the rat stayed in front of the control lever so he could press
the lever as soon as it extended typically with both hands. These movements got
stereotyped during initial sessions and stayed relatively stable afterwards.

Figure 5. Response latency to start a trial. Data from Rat O10’s learning stage III was
showed as an example. The blue box represents the range from 25 to 75 percentile, and
the red horizontal line in the middle is the median. Whiskers represent upper and lower
limits, and red crosses are outliers. This latency became larger towards the end of a
session likely due to the decreased motivation to earn food reward.

2.4 Neural modulations by choice direction

Data of 167 sessions from 8 rats (around 21 sessions per rat on average, ranging from 16
to 25) were used in this study. Each session had at least 20 L-L trials and 20 R-R trials.
Sessions with inadequate trials, mostly from early stage of learnig when behavioral
accuracy was low, were not included for neural activity analyses. Excluding these early sessions should not compromise this study because the task was learned progressively over several weeks. In contrast, animals’ behavior had not been stereotyped yet in these sessions so they were not appreciate for analysising the associative learning aspect of the task.

For each rat, only those implant sites that consistently picked up action potentials in all sessions were included. When multiple units were sorted from a single electrode in one session, the one had similar spike waveforms and firing rate as those in previous sessions was preserved. This ended up with 4.5 neurons on average (range from 3 to 6) per rat in each daily session. Therefore, the neural ensembles were comprised of the same number of units from the same subset of electrodes over sessions. Tracking single neurons in vivo is generally challenging. The analyses in this study, however, were based on neural activities at the population level which did not necessarily require same neurons over sessions. Isolated units in different sessions were therefore treated as different neurons. This left us with 747 neurons (291 from AGl and 456 from AGm) in total from all sessions of all rats.

Trials in which rats made correct directional choices were used in this study. These trials were grouped to form two classes, a first left side lever pressing in response to left side cues (L-L trials) and right side lever pressing in response to right side cues (R-R trials), in each recording session. The firing rate pattern of a single neuron in each trial was calculated as the firing rate in a 100 ms window sliding at 20 ms steps throughout the cue-on task period. A neuron’s firing activity was recognized as directional choice
modulated if there were significant firing rate differences (rank-sum test, p<0.05) between L-L and R-R trials during the cue-on period.

The neuronal mean firing rate was calculated within the cue-on task period among L-L and R-R trials, respectively. The ensemble mean firing rate was the average among all neuronal L-L and R-R rates. And the firing rate difference of the ensemble was the average among all neuronal rate differences between L-L and R-R. One session provided one sample of the ensemble mean rate and ensemble rate difference respectively. And statistics of the two rate measurements were compared over the three learning stages.

Neuronal mean firing rate dynamics of L-L and R-R trials in each session were examined, and firing rate modulations between the two classes of trials were observed (Figure 6). These modulations started to emerge after cue onset and could be found through the cue onset and conotrol lever press period, in both AGl and AGm neurons. Direction selective (DS) neurons were identified as those showing significant firing rate differences (Mann-Whitney U test, p<0.01) in at least 5 consecutive time windows (100 ms long sliding at 20 ms steps) during the cue onset period. This resulted in 51.5% neurons (385/747) being directional selective. This ratio in AGl and AGm neurons was 49.8% (145/291) and 52.6% (240/456), respectively, which was not significantly different between the two cortical areas (Mann-Whitney U test, p>0.95).
Figure 6. Raster and peri-event histograms of example neurons. Red: L-L trials in which the rat moved to the left control lever in response to left side cues. Green: R-R trials with right side movements in response to right side cues. A, example AGl neurons from three rats. B, example AGm neurons from three rats. C, AGl neurons recorded from the same site in different sessions. D, AGm neurons recorded from the same site in different sessions. Horizontal bars (black) indicate time bins in which firing rate modulation is significant (Mann-Whitney U test, p<0.01).
As a measurement of the firing rate modulation, the firing rate difference was quantized by subtracting the mean firing rate of R-R trials from that of L-L trials. In the 500 ms time window before cue onset when directional control information has not been presented, the averaged firing rate difference of all neurons is 0.01 Hz, which is not significantly different from zero (one-sample t test, p>0.93). On the contrary, within the two cue onset periods, the averaged neuronal firing rate differences are 3.00 Hz and 2.58 Hz, respectively, which are both significantly greater than zero (one-sample t test, p<0.00001; Figure 7A). The positive firing rate differences indicate the overall firing rate was higher in left side direction control trials during the cue onset period. When compared between the two time windows within cue onset period corresponding to during and after directional movements, the quantities of neuronal firing rate differences are not significantly different (paired-sample t test, p>0.22). The dynamics of firing rate difference were calculated in a sliding window manner and then averaged over neurons (Figure 7B). For the group of all neurons, firing rate difference did not emerge until 420 ms after cue onset (one-sample t test, p<0.001), and was maintained until the end of cue onset period when control levers extended. For the AGl and AGm populations, firing rate modulation reached significance (one-sample t test, p<0.001) at 320 ms and 300 ms after cue onset, respectively, and lasted through the rest of the cue onset period. These results show that firing rates of single neurons in both AGl and AGm were modulated when rats performed directional choices. Rate modulation not only presented during the execution of directional movements, but also preserved afterwards (roughly during the last 1 second of cue onset period) when there were no typical movements, implying that directional choice information was held persistently in these motor cortical neurons.
Figure 7. Dynamics of firing rate differences between L-L and R-R trials. A, Histograms of neuronal (n=747) firing rate differences (L-L subtracted R-R) in three 500 ms time windows. Upper panel: 500 ms window before cue onset. Middle panel (dark gray): 500 ms to 1000 ms after cue onset. Lower panel: 1300 ms to 1800 ms after cue onset. Vertical line: mean rate difference. B, Averaged neuronal firing rate difference in a 100 ms time window sliding at 20 ms steps from 500 ms before to 2000 ms after cue onset. Gray bars marked the time windows used in A. C, Ensemble firing rate and rate difference in a 1500 ms time window (from 300 ms to 1800 ms after cue onset) over sessions of rat B11, as an example. Upper panel: behavioral accuracy; three learning stages were defined, separated by 70% and 80% accuracy. Middle panel: mean firing rate over sessions. Lower panel: mean rate difference over sessions. Dashed lines are linear regressions. D, Statistics of ensemble firing rate and firing rate difference (6 rats, 1500 ms cue onset period). Bar: mean; whisker: standard deviation. Three learning stages were defined individually according to each rat’s learning curve.

To study how firing rate modulation in the population level changed with learning, we calculated the ensemble firing rates in each session by averaging L-L and R-R firing
rates of all simultaneously recorded neurons, and the ensemble firing rate difference was computed as the mean of neuronal firing rate differences in that session. The results using rat B11’s data was given in Figure 7C as an example, in which firing rates of a 1500 ms task period (from 300 ms to 1800 ms after cue onset) were used. Three learning stages were defined according to each animal’s learning curve, and recording sessions of different animals were then grouped according to learning stages. The ensemble firing rate did not change significantly through the three learning stages (32.37±12.60 Hz, 33.96±15.38 Hz, and 32.20±9.81 Hz; ANOVA, p>0.50) using data of the 6 rats who reached 80% behavioral accuracy. Similarly, the ensemble firing rate difference was also relatively stable over learning stages (2.68±3.39 Hz, 3.00±5.90 Hz, and 3.04±4.20 Hz; ANOVA, p>0.70). These conclusions stayed true when data of all 8 animals were used, and when firing rates were calculated in either of the two 500 ms cue onset period during and after directional movements. These findings indicate that learning directional choices did not significantly change firing rate of the recorded neural ensembles.

2.5 Conclusions

Rats learned the directional choice task from a naïve stage by trial and error. They performed stereotypical directional movements during this learning process, except in a few early sessions. Neural analyses here and below discarded early sessions without sufficient number of trials, thus, as a byproduct, excluded the motor skill learning factor. The mean firing rate at the ensemble level stayed relatively stable over learning stages. The neural modulation, i.e. firing rate differences between left and right trials, did not
change significantly either. As a conclusion, learning directional choices may not involve changes in the overall firing intensity of the neural ensembles.
CHAPTER 3 SPATIOTEMPORAL FIRING PATTERN

In the previous chapter, neural activity was measured using mean firing rates estimated in single time windows, which did not preserve the temporal firing dynamics. And neuronal firing rates were averaged within the ensemble, so the relative relation of rates of different neurons in the ensemble also disappeared. In this chapter, the neural firing activity was examined using a spatiotemporal representation which provided higher resolution both temporally and spatially.

3.1 Support vector machine decoder

We modeled the firing rate patterns of L-L and R-R trials by training linear kernel support vector machines (SVMs) to perform single-trial classification with customized Matlab programs (Mathworks Inc., MA). SVMs are supervised learning models that project training data samples to the kernel space (usually with higher dimensionality) and find the optimal hyperplane to separate data points from the two classes. The separating hyperplane is optimized so that the distance (margin) from training data points which are difficult to classify (support vectors) to the hyperplane is maximized (Burges, 1998).

When the hyperplane has been found, a SVM makes classification decision of an data sample $x$ according to the value of the decision function: $df = \sum \alpha_i k(s_i, x) + b$, where support vectors $s_i$, weights $\alpha_i$ and bias $b$ have been determined by the training process, and the function $k$ is dot product in case of a linear kernel. If $df \geq 0$, $x$ is classified as a L-L trial, otherwise it is classified as a R-R trial. The decision function value could be
interpreted as the distance in the kernel space from the sample point to the hyperplane, and the larger this distance is the less the ambiguity of classification.

A 1500 ms time period after cue onset (300 ms to 1800 ms) was used for SVM classification. This period was divided into non-overlap bins and spike counts in these bins formed one vector representation for each spike train of each neuron. Spike count vectors of simultaneously recorded neurons were then concatenated to form a neural ensemble vector representation. Thus, there was one ensemble vector as one data sample for each trial, and a SVM was trained based on data samples from two classes (L-L vs. R-R trials) in each recording session. Figure 8 illustrates the data organization and classification procedure.

Figure 8. Data preparation for SVM classification. Spike trains of all simultaneously recorded neurons of one task trial formed one data sample, where the spike counts in non-overlap bins were concatenated to form a vector representation. Two classes of data samples from L-L and R-R trials composed the data space. A SVM would transform data samples to the kernel space and find the optimal separating hyperplane.
20 L-L trials and 20 R-R trials were randomly chosen from each session. A SVM model was then trained using the first 16 L-L and 16 R-R trials, and tested by the other 4 L-L and 4 R-R trials. This procedure was repeated for 100 times for that session and the averaged testing error rate was used as the measure of SVM’s classification performance. Trial averaging could help classification performance and it was done as follows: 5 training data samples were randomly chosen from one class and averaged to create a trial-averaged sample. 480 trial-averaged samples were generated for each of the two classes and used as the training data set. To test classification performance, single-trial testing data samples were always used.

The SVM classifiers trained on one session’s data were also used to predict data from other sessions. Training data were prepared the same way as above, but testing data were 4 L-L and 4 R-R random trials from another session of the same rat. This ended up with a matrix E with the element $E_{ij}$ representing the testing error rate of SVM trained with session i and tested with session j, where sessions were arranged in chronological order. In this matrix, the diagonal elements are related to classification accuracy, e.g. $S_i = 1 - E_{ii}$ for session i. Those next to diagonal elements indicate how well a SVM classifier generalized to data of a neighboring session. Thus, the prediction accuracy of session i was defined as $P_i = 1 - E_{ij+1}$. We calculated the classifier performance and model stability for each session and focused on how the two measurements adapted over sessions with learning.
3.2 Neural pattern modeling using SVMs

With the basics about using SVMs to model neural activity being introduced above, here some variations regarding neural data preparation were tested to optimize the overall classification performance and explore characteristics of neural spatio-temporal patterns.

**Ensemble vs. single neurons.** Ensemble spike count vectors were formed by concatenating spike counts of single neurons. For each session, SVMs were trained and tested using ensemble vectors, and classification performance was characterized by test error (percentage of misclassified testing data samples). Similarly, SVMs could be built using spike count vectors of each single neuron in the ensemble. When comparing the best classification performance using single neuron data with that using ensemble data (Figure 9A), in 60.48% (101/167) sessions the ensemble outperformed the best single neuron. The mean single trial classification error among all sessions when using ensemble data was 24.86%, which is lower than 26.49% test error of the best single neuron (paired-sample t test, p<0.0001). These results suggest that directional choice information was coded coherently by neural ensembles rather than merely in isolated neurons.

**Temporal pattern vs. single time bins.** To explore the dynamics of SVM classification, a 1500 ms period during cue onset was divided into 15 non-overlap 100 ms time bins, and SVMs were trained using data of spike counts in single time bins from all simultaneously recorded neurons. As shown in Figure 9B, the classification error (averaged over all sessions) gradually decreased after cue onset and then remained relatively stable at around 40% (still lower than chance level of 50% error, one-sample t
test, p<0.00001). When spike counts in all 100 ms time bins were used for classification, the average test error is 27.67% over all sessions, which is significantly lower than single-bin cases (paired-sample t test, p<0.00001). Therefore, the temporal firing pattern, i.e. spike counts in multiple consecutive time bins, benefits SVM classification. This is consistent with the sustained firing rate modulation between left and right side choices during cue-on period (Figure 7B).

The effect of the temporal resolution, i.e. length of time bin, on SVM classification was tested. The same 1500 ms data were used but spikes were counted in bins with different lengths, ranging from 100 ms to 750ms. Best classification performance (averaged over all sessions) was obtained using 500 ms bins with 24.97% test error (Figure 9C). Longer time bins (750 ms) degraded classification performance (paired-sample t test, p<0.1) probably because of worsened temporal resolution. However, higher temporal resolutions did not benefit classification either (paired-sample t test, p<0.05). One reason might be the variation of the animal’s behavior over trials although movements were basically stereotypical. Besides, using higher temporal resolutions raise the dimensionality of data which might require extra modeling power of SVMs. Anyhow, in terms of choosing a proper data preparation format, spike counts in 500 ms bins of the 1500 ms cue onset period were used for analysis hereafter unless otherwise specified.

The histogram of classification performance in single sessions is shown in Figure 9D. The mean classification error tested with novel single trial data is 24.97%. There are sessions in which classification accuracy is as high as 90% and above, but there are also occasions when only chance level classification was achieved. Classification performance
might be improved by adopting more advanced decoding algorithms. But we did not pursue that in this study since SVM classification was used here as a means to explore the spatiotemporal neural pattern, and we focused on the adaptation of neural pattern during learning of the directional choice task.

Figure 9. Parameter set up for SVM classification. A 1500 ms cue onset period data window was used. And classifiers were always tested with single trial data. A, Classification performance using neural ensemble vs. single neuron. 500 ms non-overlap bins were used to form data vectors. Each point represents one of the 167 sessions, and those underneath the dashed line indicate higher classification accuracy when using the ensemble. B, Classification error using data of single time bins, averaged over all sessions. C, Bin size affected classification performance. 500 ms bin size and 5-trail averaging on the training data set was chosen for all following analyses. D, Histogram of single session SVM classification performance. Averaged classification error (test data set) was 24.97%.
3.3 Firing pattern adaptation with learning

To show that neural representations of L-L and R-R trials did exhibit different patterns, one run of the SVM classification was illustrated by projecting the ensemble vectors to a 2-D space by taking the first two principle components. Training data samples of the two classes formed distinct clusters and the algorithm built the separating line accordingly (Figure 10A). This classifier was then tested with novel single trial data samples, and the averaged decision function value (see section 3.1) of testing samples from either class was shown as a line parallel to the separating line (Figure 10B). The margins between those lines provide an intuitive measurement of the separability of the two classes, in addition to the classification accuracy.

For each session, we ran the classification for 100 times by randomly selecting training and testing samples. Figure 10C showed the histograms of decision function values of testing samples from all runs in two sessions from rat B11 as an example. Decision function values from the two classes were significantly different (ANOVA, p<0.00001) in both sessions. But the distance between the centers was larger in the later session (2.22 vs. 0.77) when SVM classification was also better (test error 24.50% vs. 42.25%).
Figure 10. Classification performance improved over learning stages. A, An illustration of training data samples, support vectors and separating plane (black line) in a 2-D data space. Data of rat B11 session 19 were used, and the dimension of data samples was reduced to 2 for display by taking the first two principal components. B, The original single trials used to generate trial-averaged training data and single-trial test data samples plotted in the same space as in A. Colored lines were parallel to the separating line with distances equal to the mean distance of test samples to the line. C, Histograms of test sample decision function values of session 11 and 19 of rat B11. D, An example of classification performance over sessions using data of rat B11. Upper panel: Classification error tended to decrease with learning. Lower panel: the distance between L-L and R-R data sets in the SVM kernel space increased with learning. E, Classification error decreased and distance between two classes in SVM kernel space increased over three learning stages (6 rats, 1500 ms cue onset period). The 3 learning stages were defined the same way as in Figure 7D.
When compared over sessions, the test error showed a decreasing trend, and the distance between centers of the two classes (mean decision function values of test samples) increased with learning (Figure 10D, rat B11 as an example). To summarize results from different animals, recording sessions were divided into three learning stages in the same way as in Figure 10D. Among the 6 rats who reached 80% behavioral accuracy, classification error decreased from 32.82% in stage I to 27.88% in stage II, and finally reached significance with 22.82% error in stage III (ANOVA, p<0.001; Figure 10E). The center distance increased significantly with learning, from 1.42, 1.73, to 2.07 (ANOVA, p<0.005; Figure 10E). These results revealed enhanced discrimination of neural spatiotemporal firing patterns between left and right side choices as learning progressed.

SVM classifiers trained on one session were also tested with data from other sessions of the same animal (Figure 11A). Low classification errors were achieved generally in sessions around the diagonal line, indicating relatively stable neural firing patterns among neighboring sessions. Both classification and prediction accuracy (see section 3.1) showed increasing trends as learning progressed (Figure 11B) for the six rats who achieved 80% behavioral accuracy or above. These results further supported the conclusion that neural spatiotemporal firing patterns adapted to exhibit enhanced representation of choice direction.

As the cue-on period has been of major interest throughout the study, the control lever press period is also examined as a control. A 500 ms time window (from -100 ms to 400 ms) around control lever press (CP) was used, and spike counts in 100 ms non-
overlap bins were used to build SVMs to decode left side vs. right side lever press.
Classification accuracy also exhibited a rising trend (1.00 slope of linear regression, same method as in Figure 11B) using control lever press data, but it is not as steep as classification performance improvements in the two time windows from cue-on period (1.77 and 1.95 slope of linear regression in CO1 and CO2 window respectively; rank-sum test, p<0.01). Similar results were observed for the prediction accuracy measure (0.37 slope of linear regression in CP vs. 0.54 in CO1 and 1.54 in CO2; rank-sum test, p<0.01).
When calculated as in the three learning stages (Figure 11C), classification performance did not change significantly over stages for control lever press period (test error 30.76%, 24.73%, and 26.74%; ANOVA, p>0.15 between stage I and III). For the two cue onset periods, in contrast, classification performance consistently improved through the three stages (CO1: 40.16%, 38.88%, and 35.59%; CO2: 40.81%, 34.42%, and 30.06%; ANOVA, p<0.05 between stage I and III). Taken together, enhanced discrimination of neural firing patterns presented during the cue-on period when animals were learning to make directional choices, but not when they were executing control lever press which was a motor skill they already learned.
Figure 11. SVM classification performance and prediction accuracy. (A) SVM classifiers trained with one session (the model session) and tested with another (the test session). Testing error rates (%) were color coded. Rows and columns were arranged from the first to the last session. Results from all 8 rats were presented individually. The bottom right corner matrix illustrates classification performance as in a diagonal element (blue) and prediction performance as in the element (orange) to the right of the diagonal. (B) Both classifier performance and model stability increased with learning, using 6 rats whose behavioral accuracy had reached 80%. (C) Classification performance improved over the three learning stages in the two cue-on period data windows (CO1 and CO2) but not in the control lever press data window (CP).
3.4 Discussions

Rats were conditioned to respond to either a left side or a right side lever according to the position of the light cue. They made directional choices during the cue-on task period by moving to either side and got prepared to press the control lever once it extended. The directional movements got stereotyped within the first few days so the motor skill learning factor could be excluded from the rest of the associative learning process. During this process, the neuron averaged mean rate each day did not change significantly among the three learning stages. The difference in mean rates between left and right side trials did not change significantly either. When using SVMs to decode choice direction from spatiotemporal neural firing patterns, improved decoding accuracy indicated enhanced discriminability between neural patterns during left side and right side choices. These findings suggest that neural adaptation in rat AGm and AGl areas during learning of the directional choice task lies in the spatiotemporal firing pattern of neural ensembles. And it might not involve mean rate changes as long as movement parameters remained stable.

**Firing pattern adaptation during learning**. SVM classifiers were constructed not to seek for a neural decoder, but to measure the discriminability of neural firing patterns. In this sense, consistent classification power was used to examine data from different sessions. First, the same linear kernel SVM model was used for all analyses. Second, each and every classifier was trained with samples generated from the same number of 32 trials, and tested with data of 8 trials. Hence the changes of SVM classification
performance over sessions tended to reflect the characteristics of neural firing patterns which adapted with learning.

Only correct trials were used in neural activity analyses, so trials of the same class, either L-L or R-R, from different sessions were identical in terms of cue light direction and animal’s choice direction. The response time analysis (Figure 3B) indicated that motor skill learning presented only in the first 2-3 days, and those sessions were not included in neural activity analyses due to insufficient number of correct trials. Although there could be some detailed kinematic changes during the rest of the learning process, the directional movements we monitored remained stereotypical without any systematic changes observed. Therefore, the main factor that induced the observed neural adaptation would be learning of the directional choice task: from making choices arbitrarily by trial and error to following the associations between cue positions and choice directions.

Could the changes in neural firing patterns come from repetition of the directional movements? Using neural activity during control lever press as a control, classification performance did not change significantly from learning stage I to stage III (Figure 11D). Control lever press was also an action repeatedly executed during learning of the task, but it was a familiar motor skill that animals already learned before working on the directional choice task. Previous primate studies show that mere repetition of familiar actions did not induce systematic changes in motor neurons (Paz and Vaadia, 2004; Rokni et al., 2007). It’s also worth noticing that control lever press was performed after a directional choice was made. By the time control levers extended, rat was already in front of one control lever while the other was out of his reach so there would be no ambiguity
which lever to press at that time. These facts further support the conclusion that the adaptation of neural pattern could be related to the change of how directional choices were made throughout the learning process.

In this work, we treated single units recorded from the same electrode in different sessions as independent samples. We had no intention to track or compare activity of single neurons over sessions. Instead, results were based on statistics of some measure in a group of sessions in a learning stage. Considering that picking up a new unit would likely have a random effect on the measure observed from the old unit, any statistically significant change of that measure would be unlikely due to changing units. Therefore, the consistent and significant enhancement of discriminability of firing patterns over the three learning stages would not be compromised by whether we recorded the same neurons over sessions or not.

**Rat AGm and AGl areas.** The recording sites were decided according to rat brain atlas (Paxinos and Watson, 2005) and confirmed by stimulations. Whisker movements were evoked by the minimum intensity of electrical stimulation (60 μA) through electrodes in AGl but not in AGm (Figure 1C). Rat AGl has been considered to correspond to the primary motor (M1) cortex (Donoghue and Wise, 1982; Donoghue and Parham, 1983), while the homologue of rat AGm in primates is still under debate.

The term AGm refers to the medial subdivision of the agranular field of rat frontal cortex which differs from the lateral subdivision (AGl) on cytoarchitectonic grounds (Donoghue and Wise, 1982). Other terms referring to this area used in literature include medial precentral area (PrCm, Krettek and Price, 1977), frontal cortical area 2 (Fr2, Zilles
1985), and secondary motor area (M2, Paxinos and Watson, 2005; MOs, Swanson 1998). Besides the inconsistent nomenclature, although rat AGm has been proposed to be homologous to premotor cortex, supplementary motor area, and frontal and supplementary eye fields in primates (Donoghue and Wise, 1982; Reep et al., 1987; Van Eden et al., 1992; Condé et al., 1995; Preuss, 1995), there is a different view that rat AGm has dorsolateral-prefrontal-like features both anatomically and functionally (Uylings et al., 2003). The arguments about the homologue of rat AGm in primates are beyond the scope of this study. Besides, both premotor and prefrontal characteristics may coincide in some architectonically less differentiated frontal areas in the rat brain.

In previous neuropsychological studies, lesions of AGm impaired both the retrieval (Passingham et al., 1988) and the acquisition (Winocur and Eskes, 1998) of visuomotor conditioning, which suggest a role for AGm in stimulus-response associative learning. In line with these reports, we observed firing pattern adaptation during associative learning within the motor cortical neural ensembles, which consisted of AGm neurons and AGl neurons near the AGl-AGm border. It is worth noticing that AGm is a cytoarchitectonic region which may have functionally distinct subdivisions. Neurons we recorded tended to congregate towards the middle part of the rostro-caudally arranged AGm area. In another study, neural activity related to value-based action selection was observed in rostral AGm which was compared to primate SMA (Sul et al., 2011). Therefore, future studies in more confined regions within AGm or AGl under the same directional choice learning protocol would help to testify whether the observed adaptation originated in a specific region or is wide spread in the AGm and AGl area.
Spatio-temporal firing patterns. As mean firing rate is the most common metric in firing activity analysis, spatiotemporal neural activities have been used to study various cortical functions as well, such as visual attention (Heinze et al., 1994), odor representation (Laurent et al., 1996; Spors and Grinvald, 2002; Rennaker et al., 2007), auditory processing (Kayser et al., 2009) and vibrissa deflection coding (Petersen and Diamond, 2000), to name a few.

There were previous studies investing the occurrence of precise firing sequences using sensorimotor tasks (Abeles et al., 1993; Prut et al., 1998). Here we calculated spatiotemporal patterns of firing rate in time bins of hundreds of milliseconds, and focused on their adaptation with learning. In a rat reaction-time task, fine temporal (10 ms or less bins) firing rate patterns of motor cortical ensembles predicted single trial outcome better with training (Laubach et al., 2000). The use of a relatively subtle and well aligned action, lever release, was probably the reason that prediction optimized when using fine time bins. In their study, 25.4 neurons per session were obtained and about 79% classification accuracy was achieved in fully trained animals. In this study, comparable single-trial classification accuracy (75% over all sessions) was obtained with smaller ensembles (4.5 neurons per session).

In a similar directional choice task (Cohen and Nicolelis, 2004), prediction of movement direction by M1 ensemble improved during initial days while motor skill was learned. However, results in later sessions as stimulus-response association learning progressed were unfortunately not available. Besides, increased firing rate difference between left and right movement was observed only during the first day when motor skill
learning dominated. Similarly, we didn’t observe changes in rate difference between left and right choices probably due to animal’s directional movements already stereotyped. In another study (Huber et al., 2012), motor parameters were decoded by layer 2/3 M1 neural activities imaged every 250 ms. Neural population representation of the response action, licking, strengthened during learning, but that of the sensing action, whisking, remained stable. In our study, enhanced firing pattern was found when directional choice was made in response to light cue, in layer 5 neurons from AGm and AGl area. Collectively, it is suggested that motor cortical neuronal firing rate changes would emerge during motor skill learning, while executing stereotypical movements in different contexts involves tuning of the spatiotemporal firing pattern of motor cortical neural ensembles.

**Neural plasticity with learning.** In this experiment, animals learned to choice proper response with the goal to earn food reward. The reward-related decision making process is believed to be mediated by the corticostriatal circuitry, linking prefrontal, premotor, sensorimotor cortices and the striatum (Balleine et al., 2007). In an associative learning study, Pasupathy and Miller (2005) found striatum showed rapid changes compared with a slower trend in PFC, suggesting that rewarded associations were first identified by the basal ganglia and then influenced PFC. In the rat brain, both AGm and AGl project to basal ganglia (Reep et al., 1987; Cheatwood et al., 2003; Alloway et al., 2009), and both areas receive inputs from basal ganglia through the relay of thalamus (Donoghue and Parham, 1983; Reep et al., 1984). So the firing pattern adaptation in AGl and AGm ensemble observed here could be mediated by this neural network when rewarded directional choices were learned.
Rat motor cortex is highly capable of functional and structural changes even in the adulthood. Reorganization of motor maps has been observed in various experiments (Sanes et al., 1990, 1992; Lee et al., 2003), including animals learning a motor skill (Nudo et al., 1996; Kleim et al., 1998, 2004). Cortical synaptogenesis has been reported during motor training (Jones et al., 1999; Kleim et al., 2004). And recent studies demonstrated learning-induced dendritic spine changes in rodents performing motor tasks (Xu et al., 2009; Yang et al., 2009; Wang et al., 2011). While these changes were related to learning of certain motor skill, whether learning stimulus-response association would induce such changes in the motor cortex is unclear. Synaptic plasticity has long been hypothesized for being an important neurochemical foundation of learning and memory (Malenka and Bear, 2004; Gilson et al., 2010). While its necessity has been well supported, the notion of sufficiency is barely tested (Martin et al., 2000). In our experiment, the directional choice task took animals several weeks to learn. Such a long learning process would possibly allow synaptic modifications to occur. And as a result, enhanced spatiotemporal neural representations of directional choices developed with learning as rewarded choices were selected more accurately.
CHAPTER 4 FIRING VARIABILITY ANALYSIS

Neural firing activity is highly stochastic, and it varies even in repeated trials with highly consistent external parameters. As the mean firing rate among repeated trials has been widely used to measure firing activity, the trial-to-trial firing variability serves as an additional measurement, which could also provide information about neuron’s engagement when performing behavioral tasks. A decline in trial-to-trial firing variability was observed after stimulus onset in multiple datasets recorded from different cortical areas (Churchland et al., 2010). And dynamics of firing variability presented in and correlated with other behavioral tasks (Churchland et al., 2006; Hussar and Pasternak, 2010) as well. In a motor skill learning task when rats learning reach-to-grasp, the variability significantly decreased with training (Kargo and Nitz, 2004). So here neuronal trial-to-trial firing variability was examined to search for any relevance to learning the directional choice task.

4.1 Trial-to-trial firing variability during learning

As shown in previous chapters, the spatio-temporal firing patterns of L-L and R-R trials became more separable using linear SVM classifiers as learning progressed. And at the same time, the mean firing rate difference between the two types of trials remained relatively stable over the three learning stages. In other words, improved discriminability of the spatiotemporal firing pattern was not due to increased mean firing rate difference between left and right trials. Another possible way to achieve the increased firing pattern discriminability is to restrain the trial-to-trial firing variability, such that firing patterns of
trials in the same class become more similar to each other so there is less ambiguity between the two classes. Separability is improved in this case without an increase of the distance between the two.

To test this hypothesis, the Fano factor (FF, spike count variance over spike count mean) values of single neurons were calculated and compared over the three learning stages. The CO1 data window (0.5 s – 1.0 s after cue onset) was used here since directional choices were characterized by stereotypical movements in this task period. Spike counts in this task window were calculated in single trials. For each recorded unit in one session, FF was computed for L-L and R-R trials respectively using these spike counts. Neuronal FF values were then compared as in the three learning stages. As shown in Figure 12, this neuronal FF of spike counts did not change significantly (p>0.5, both Mann-Whitney U-test and Kolmogorov-Smirnov test) between stage I and stage III, in either L-L trials or R-R trials. The same conclusion could be made in CO2 (1.3 s – 1.8 s after cue onset) and CP (-0.1 s – 0.4 s around control lever press) task windows as well. Thus, the trial-to-trial firing variability as measured by FF of spike counts did not decrease over the three learning stages.

Reduced FF was observed in M1 neurons in a previous reach-to-grasp learning task (Kargo and Nitz, 2004). In a study using a similar directional choice task (Cohen and Nicolelis, 2004), firing variability did not change during learning of the motor skill of directional movements. In the current experiment, the motor skill of lever pressing was already learned before rats started working on the choice task, and learning directional movements completed within about the first 3 days most of which were excluded from
neural analyses due to insufficient number of correct trials. Therefore, motor skill learning was unlikely a factor in the current study. Thus, even if improving motor skills would cause decreased firing variability, it is not expected to be observed in this study during learning of directional choice. These results also suggest that neuronal firing rate variability is not likely the neural substrate of the enhanced spatiotemporal firing patterns.

Figure 12. Statistics of neuronal Fano factor values of spike counts in the three learning stages. Left: L-L trials. Right: R-R trials. The blue box represents the range from 25 to 75 percentile, and the red horizontal line in the middle is the median. Whiskers represent upper and lower limits, and red crosses are outliers. FF values were not significantly different between learning stage I and III for either L-L and R-R trials.

4.2 Dynamics of firing variability according to task demands

Trial-to-trial firing rate variability (FF) was also calculated in a time-resolved manner (using 100 ms sliding window moving at 20 ms steps; all trials in each session combined), and the averaged FF of all single neurons was plotted in Figure 13A. The averaged neuronal FF showed a declining trend from 1 sec before to 2 sec after cue onset.
Neuronal FF was significantly smaller in the two post-cue windows, CO1 and CO2, compared with a pre-cue period (Mann-Whitney U-test, p<0.005; Figure 13B).

Figure 13. Dynamics of firing variability in the period of one trial. FF decreased after cue onset.

Notice that all trials, both L-L and R-R, were grouped together to calculate the neuronal FF in this analysis. So the firing variability in CO1 and CO2 periods should also include the variation between left and right side directional movements. Even so, FF in the Pre task window, when rats reached to the center ready lever, was still higher than in cue-on windows. This declination in firing variability could be due to increased task demands after cue onset: After a cue light was on, animals observed the cue, chose one direction, and started to move towards the control lever in that direction as quickly as possible. Increased cognitive demands to make directional choice decisions, together with other possible task related factors, might be correlated with the observed variability reduction after cue onset.
Several task related factors were actually coincident during the cue-on task period. One would be the presentation of the visual stimulus starting from ready lever press. The LED cue lights continuously provided indication regarding animal’s progress towards the goal, to move the cue light to the center position to receive a sugar pellet, until the end of a trial when a tone was played according to the outcome. The trial-to-trial spike count variability declined after cue onset and started to rise after trial ended (Figure 13A). The time course of the reduced variability matched the duration of the visual cues. Besides, attention could be another factor being involved during the period of performing a trial. Movement execution appeared to modulate the variability as well since it showed a decrease right around control lever press (Figure 13A). Currently it is difficult to dissociate these factors, and they might all contribute to the decreased firing variability when performing the task.

4.3 Firing variability of two distinct task factors

By far, trials from the same session were grouped as L-L and R-R trials to study the directional neural modulation of AGm and AGl neurons. As animals learned the directional choice task by trial and error, trial outcomes were also crucial information that might be coded in these cortical areas. The same group of trials in each session were then divided into two groups as following either a successful (P-S, post-success) or a failed (P-E, post-error) previous trial. Previous outcome selectivity (POS) was defined as firing rate modulation between P-S and P-E trials, similar as direction selectivity (DS) defined between L-L and R-R trials.
Some example neurons were shown in Figure 14, where neurons could be DS (Figure 14A), POS (Figure 14B), or showing both types of selectivity (Figure 14C). The number of neurons showing DS or POS was counted in a time-resolved manner through the entire task period (Figure 14D). DS neurons started to emerge after cue onset, while POS neurons presented before trial started and sustained most parts of the task duration. The number of DS neurons was mostly stable in the cue-on period, but the number of POS neurons was initially high after cue onset, and decreased as directional choice was made and executed. When the entire task period was considered together, around 80% neurons exhibited DS in at least one time bin (100 ms). About half of these neurons showed DS but never POS. On the contrary, a large portion of POS neurons also showed DS in some other task periods. In summary, both direction information and previous trial outcome were coded in these motor cortical neurons. When trials were grouped as L-L and R-R, as in most of the results shown in previous chapters, some firing rate variation had been introduced into both groups due to the existence of POS neurons.
Figure 14. Both task factors, choice direction and previous trial outcome, were coded in motor cortical neurons. A, An example neuron showing DS but not POS. B, An example neuron showing POS but not DS. C, An example neuron showing both types of selectivity. D, Total number of neurons showing DS (black) or POS (gray) through the task period. E, Percentage of neuron ever showing selectivity within the whole task period.

We then calculated the spike count variability of DS and POS neurons separately. For DS neurons, trials were grouped as L-L or R-R, and for POS neurons, trials were grouped as P-S and P-E. The CO1 task period was used to identify DS and POS neurons, and this ended up with 160 DS neurons and 203 POS neurons from all animals (n=8). The temporal dynamics of FF of the four trial groups were shown in Figure 15A. The FF
values of P-S and P-E trials of POS neurons were significantly higher (Mann-Whitney U-test, p<0.001) than those of L-L and R-R groups of DS neurons. When these FF values were divided into the three learning stages, none of the four trial groups changed their FF significantly between stage I and stage III (Figure 15B; Mann-Whitney U-test, p>0.1).

![Figure 15](image)

**Figure 15.** Direction selective neurons exhibited smaller variability than previous outcome selective neurons.

The presence of both DS and POS indicated that neuronal firing rate could be influenced by various task-related factors, and individual neurons might not just play a single constant role but might work cooperatively within a neural network in order to fulfill complex cognitive functions, such as learning the directional choice task. The relatively stable FF values of DS and POS neurons in the three learning stages supported
previous results (Figure 13B) that single neurons did not fire with reduced trial-to-trial variability with learning.

4.4 Trial-to-trial firing rate fluctuation

Another observation from the neural data was that the overall firing rate within one trial’s duration could be very different from trial to trial. The shift of the baseline firing rate was considered as the cross-trial fluctuation. As there was no single task period could be used as the baseline (animals were always free to move), the mean firing rate in the period from 1 sec before to 2 sec after cue onset was calculated to represent the overall firing rate of each single trial. Differences of the overall firing rate among trials from the same group (L-L, R-R, P-S, or P-E in a single session) were then removed from all single trials. FF was recalculated in the same time-resolved manner after this procedure.

Figure 16. Trial-to-trial firing rate fluctuation explained part of the variation.
As shown in Figure 16, FF values decreased for all four groups of trials after baseline fluctuations were removed. This confirmed that cross-trial firing rate fluctuation did present in the recorded neural data and could explain part of the observed firing variability. Interestingly, other characteristics of the trial-to-trial firing variability were preserved after fluctuation removal, such as decreased variability after cue onset and higher variability of POS neurons than DS neurons. These indicated that cross-trial firing rate fluctuation could be originated from a mechanism different from that of the other two observed phenomena. And again, neural activity seemed to be influenced by complex compounding factors which might make firing patterns highly stochastic and challenging to analyze and interpret.

4.5 Summary

Neuronal trial-to-trial firing variability was examined in this chapter, and it did not change significantly with learning over the three stages. Previous studies showed that learning a motor skill would induce a declination in trial-to-trial firing variability. This effect was not observed in this experiment because animals were learning the associations between visual cue and directional choice without any significant tuning of their directional movements. Thus, the improved discrimination of neural spatiotemporal firing pattern could not be correlated with either the mean firing intensity of the neural ensemble during directional choice or neuronal trial-to-trial firing variability.

So far, neural firing activity was modeled separately as in individual sessions. Firing patterns of L-L and R-R trials in one session adapted to show increased
discriminability with learning. But whether neural patterns of the same type of trials in different sessions shared certain common features is unclear. In the next chapter, models were built to capture features of firing patterns from data recorded in all sessions. And advanced machine learning tools were adopted for such purpose.
CHAPTER 5 MODELING NEURAL FIRING PATTERNS THROUGHOUT THE LEARNING PROCESS

The spatiotemporal neural firing patterns are complex signals. Their dimensionality could be very high with a fine temporal resolution and a large neural ensemble size. Neuronal firing activity is usually modeled as a stochastic process and so are these firing patterns. On top of these, during a learning process, the firing pattern may undergo systematic adaptations over time. Not to mention that various sources of neural noise and issues of single-unit recording may introduce additional variations or irregularity into the observed firing patterns. All these factors make analyzing spatiotemporal firing pattern and its adaptation a challenging task. Recent advances in the machine learning community, specifically deep learning algorithms, have demonstrated powerful tools in modeling patterns from massive data sets. Here we adopted some of those techniques to further analyze and explore the motor cortical neural firing patterns as animals learned the directional choice task.

5.1 Energy-based models and restricted Boltzmann machines

Energy-based models are statistical modeling tools which encode dependencies between variables (LeCun et al., 2006). Such models could include two sets of variables: observed variables $x$, for example, the neuronal firing rates of a group of neurons; and hidden variables $h$, for example, animal’s directional choice or other factors which are not readily available in the neural signal. The model is represented by the energy function
The energy of a configuration of \( x \) and \( h \) would be turned into the probability of observing this configuration following this equation, \( P(x, h) = e^{-E(x,h)} / Z \), where

\[
Z = \sum_{x,h} e^{-E(x,h)}
\]

is the normalization term called the partition function. In unsupervised learning cases, the principle used to build the model would be to describe the distribution of \( x \), i.e., to maximize \( P(x) \). Given the form of the energy function (a function of \( x \) and \( h \) with parameter \( \theta \)) and observed values of \( x \), a model parameter \( \theta \) could be determined so as to maximize the log-likelihood of observing those \( x \) values. The gradient of the log likelihood with respect to parameter \( \theta \) can be written as follows:

\[
\frac{\partial \log P(x)}{\partial \theta} = \frac{1}{P(x)} \frac{\partial P(x)}{\partial \theta} = -\sum_h P(h|x) \frac{\partial E(x,h)}{\partial \theta} + \sum_{\tilde{x},h} P(\tilde{x}, h) \frac{\partial E(\tilde{x}, h)}{\partial \theta}
\]

In this equation, \( x \) represents observed data used to train the model, and \( \tilde{x} \) is a possible value of observed variables from that model. Given the gradient of the log likelihood function, model parameter could be optimized according to gradient descent methods.

There are two expectations of the gradient of energy function in the above equation: the first is among all \( h \) values given observed \( x \), the second is among all possible \( \tilde{x} \) and \( h \) configurations according to the model. The gradient of the energy function is easy to compute once the form of the function has been decided. For a given observed value of \( x \),
$P(h|x)$ is also computable given this conditional distribution takes certain form according to the model. What is challenging is to find $P(\bar{x}, h)$ or to sample $\bar{x}$ and $h$ from the model.

A useful tool to sample $P(\bar{x}, h)$ would be Gibbs sampling (Casella and George, 1992), which is a Markov chain Monte Carlo algorithm to generate a sequence of samples that are approximated from the joint probability distribution. The key of Gibbs sampling is to use the conditional distribution to draw a random sample of one variable given a set of values of all other variables. Suppose we want to obtain a sample of $S = (s_1, ..., s_n)$ from the joint distribution $P(s_1, ..., s_n)$. We begin with some initial value $S^{(0)} = (s_1^{(0)}, ..., s_n^{(0)})$, and sample a value $s_i^{(1)}$ for variable $s_i$ from $P(s_i|s_1 = s_1^{(1)}, ..., s_{i-1}^{(1)}, s_{i+1} = s_{i+1}^{(0)}, ..., s_n = s_n^{(0)})$. We do this sequentially for all $n$ variables and then obtain a new sample $S^{(1)} = (s_1^{(1)}, ..., s_n^{(1)})$. Then we repeat this procedure to get $S^{(2)}$ based on $S^{(1)}$, and keep repeating to get the $k$-th sample $S^{(k)}$. These samples, $S^{(0)}, S^{(1)}, ..., S^{(k)}$, form a Markov chain, and the distribution of samples in the chain would converge to $P(s_1, ..., s_n)$ when $k$ goes to infinite.

Running a long Markov chain is very time consuming. But if we could initiate the chain using samples from a distribution close to $P(\bar{x}, h)$, then the chain doesn’t have to be long to converge. The empirical distribution of $x$ from observed values (data used to train the model) is a good starting point in this sense, and then we only run the chain for a finite number of $k$ steps to get a sample approximately from the model itself. This method is named $k$-step contrastive divergence (CD-$k$). Although it is a second approximation on top of the Gibbs chain, it has been shown to be useful to estimate the
gradient of the log likelihood function in models such as restricted Boltzmann machines (Bengio and Delalleau, 2009).

![Illustration of a restricted Boltzmann machine (RBM) and a deep belief network (DBN).](image)

Figure 17. Illustration of a restricted Boltzmann machine (RBM) and a deep belief network (DBN).

A restricted Boltzmann machine (RBM) is an energy-based probabilistic model which associates with the following energy function

$$E(x, h) = -b'x - c'h - h'Wx,$$  \hspace{1cm} (5.2)

where $b = (b_1, ..., b_j, ..., b_N)'$, $x = (x_1, ..., x_j, ..., x_N)'$, $c = (c_1, ..., c_j, ..., c_M)'$, $h = (h_1, ..., h_i, ..., h_M)'$, $W$ is a $M \times N$ matrix with elements $w_{ij}$, and $M$ and $N$ are the number of variables in $h$ and $x$, respectively. In this model, $b_j$ is the bias of visible variable $x_j$, $c_i$ is the bias of hidden variable $h_i$, and $w_{ij}$ represents the interaction between $h_i$ and $x_j$. And they are the parameters of the model need to be decided from data. From a network point of view as illustrated in Figure 17A, connections between units (nodes) only exist between the two layers, an input (visible) layer and a hidden layer, but not between units.
in the same layer. In generic RBM models using the above energy function, both \( h_i \) and \( x_j \) are binary (take 0 or 1 values). For continuous-valued visible variables, such as neuronal firing rates in our case, extensions from the binary case could be made.

In case of Gaussian \( x_j \) and binary \( h_i \), the energy function could take the following form,

\[
E(x, h) = \sum_j \frac{(x_j - b_j)^2}{2\sigma_j^2} - \sum_i c_i h_i - \sum_{i,j} h_i w_{ij} \frac{x_j}{\sigma_j}.
\] (5.3)

Its partial derivatives with respect to model parameters are straightforward to compute,

\[
\frac{\partial E(x, h)}{\partial \omega_{ij}} = -h_i \frac{x_j}{\sigma_j},
\] (5.4)

\[
\frac{\partial E(x, h)}{\partial c_i} = -h_i,
\] (5.5)

\[
\frac{\partial E(x, h)}{\partial b_j} = -\frac{x_j - b_j}{\sigma_j^2},
\] (5.6)

\[
\frac{\partial E(x, h)}{\partial \sigma_j} = -\frac{(x_j - b_j)^2}{\sigma_j^3} + \sum_i h_i w_{ij} \frac{x_j}{\sigma_j}.
\] (5.7)

And the conditional distributions are,

\[
P(h_i = 1|x) = \text{sigm}(c_i + \sum_j w_{ij} \frac{x_j}{\sigma_j}).
\] (5.8)

\[
P(x_j|h) = \mathcal{N}(b_j + \sigma_j \sum_i h_i w_{ij}, \sigma_j^2),
\] (5.9)

where \( \text{sigm}(t) = \frac{1}{1 + e^{-t}} \), and \( \mathcal{N} \) denotes Gaussian distribution. Thus, the gradient of log likelihood of \( P(x) \) with respect to model parameters could be calculated by substituting equations (5.4) ~ (5.9) into equation (5.1), and model parameters could be optimized accordingly to maximize the likelihood of observing \( x \).
Specifically when calculating the gradients of log likelihood given data $x^{(0)}$, the first term in equation (5.1) is based on the input data so should be computed with $x = x^{(0)}$ and $h^{(0)}$, where $h^{(0)}$ is sampled from $P(h^{(0)} = 1|x^{(0)})$ using equation (5.8). To compute the second term of equation (5.1), according to Gibbs sampling and CD-$k$ method, we run a Markov chain $x^{(0)} \rightarrow h^{(0)} \rightarrow x^{(1)} \rightarrow h^{(1)} \rightarrow \cdots \rightarrow x^{(k)} \rightarrow h^{(k)}$ following equation (5.8) and (5.9), and take $P(x^{(k)}, h^{(k)})$ as the joint probability of the variables according to the model. In practice, CD-1 is often used which is computationally feasible and could train good models in practice. It may not compute accurate estimations of the gradient of the log likelihood, but at least follows the sign of the gradient generally correctly (Bengio and Delalleau, 2009), which ends up with adjusting model parameters in the correct direction.

As illustrated in Figure 17B, stacking multiple layers of stochastic variables could form the so called deep belief networks (DBNs). DBNs are probabilistic models which could be trained in an unsupervised, layer-to-layer procedure to learn a hierarchical representation of the training data (Hinton and Salakhutdinov, 2006). When training a DBN, we go through the network layer by layer using a bottom-up approach and treat each pair of two adjacent layers (e.g. $x \leftrightarrow h^1$, $h^1 \leftrightarrow h^2$) as an RBM model. Above described RBM training methods would be applied to train each hidden layer in the DBN. After all layers in the DBN have been trained, the network would have been initialized as a feature extractor customized for the training data, and additional layers could be added on top of the DBN to perform tasks such as pattern recognition and classification which make use of extracted features by the DBN. Any following supervised training procedure
to fine tune the entire network would benefit from the properly initialized DBN, so it’s more feasible to train a deep network in this way compared with directly training a totally randomly initialized network just using back propagation.

5.2 Neural firing pattern modeling using RBMs

To train an RBM model using neural data, two considerations went into the organization of neural data. Computationally, training an RBM needs a large data set which could be made up of data from multiple recording sessions. Even though the electrode may pick up different neurons over sessions, we still consider the same electrodes would record from a neural ensemble of similar statistical properties. Furthermore, given that animal’s physical movement usually became stereotypical pretty quickly, we assume that within a single learning stage, there exists emerging neural activity patterns and they may be different as learning stage changes from the first to the second and then to the third.

Data preparation. For each animal, all sessions from all three learning stages throughout the learning process were included as one dataset as long as there were at least 20 correct trials in both directions (left and right). We randomly chose 20 trials from each of the two classes, and kept 4 of them (8 single-trial samples in total) as testing data samples. Among the rest 16 trials, trial-averaged data samples were created as the mean firing pattern of 5 random trials. 500 trial-averaged data from each class were obtained, and 450 of them were used for training and the rest 50 for validation. Data samples of single sessions were then grouped to form the training, validation and testing sets respectively, and together they formed one complete data set. The above procedures were
repeated 100 times to create 100 data sets using randomly selected trials of the same animal. Each data set was used to build one RBM model, and statistics regarding RBM modeling could then be obtained from these 100 models.

To form a single trial neural firing pattern, a 1500 ms data window within the cue onset period was used (from 300 ms to 1800 ms after cue onset). The neuronal spike counts in 15 non-overlap 100 ms time bins of this window were computed, and spike counts of all simultaneously recorded neurons were concatenated to form one spike count vector (in a similar way as shown in Figure 8 in chapter 3). For example, for rat A09, there were four isolated neurons each from a different electrode, so the single trial firing pattern data would be a $1 \times 60$ vector which was comprised of four $1 \times 15$ spike count vectors of the four neurons. All data samples from each session were normalized respectively according to the mean and standard deviation of spike counts in one time bin.

*Training RBMs.* To train an RBM, all training samples from both left and right classes and from all sessions were used to estimate the gradient in equation (5.1) with CD-1. Model parameters were then adjusted along the direction of the gradient with a typical learning rate of 0.01. This procedure was repeated once in each learning epoch until the mean reconstruction error between $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$ as in the chain $\mathbf{x}^{(0)} \rightarrow \mathbf{h}^{(0)} \rightarrow \mathbf{x}^{(1)}$ stopped declining. Typical learning curves when training an RBM are shown in Figure 18 using data from animal W09. It usually took 100 to 200 epochs for the model to converge. Adding more hidden units in the RBM reduced the reconstruction error, but the effect became marginal when using more than 50 units. Using larger size models may introduce overfitting to the model and thus results in poor generalization.
Figure 18. Example learning curves of RBM training.

*Evaluation of features extracted by RBM.* Data fed into the RBMs are spatiotemporal neural firing patterns. What RBMs extracted could be some abstract features in those patterns that would depend on recorded neurons so might change from animal to animal, from neural population to neural population, and from task period to task period. Instead of digging into the details of the extracted features, we examined whether and how they encoded task related information, and in our case the directional choice information.

To test this, we took the features of training data samples extracted by the RBM to train a SVM classifier and decode the direction of animal’s choice. Features of the single trial testing samples were then used to test SVM’s classification performance ($SVM_h$ in Figure 19). As a comparison, another SVM classifier was trained and tested using the original firing pattern data ($SVM_x$ in Figure 19). We hypothesize that the classification performance using RBM processed data would be better than using raw spike count data,
which indicates the effectiveness of RBM in extracting directional choice related information from neural activities.

![Diagram](image)

Figure 19. DBNs extracted features beneficial to direction information decoding.

Data from nine animals learning the directional choice task was used here. First we tested how the size of the RBM affected decoding performance. 100 models of SVM classification based on RBM extracted neural features were trained for each animal, and this was repeated with RBMs with the number of hidden units ranging from 5 to 100. As shown in Figure 20, classification performance improved as hidden unit number in RBM increased from 5 until it reached a plateau when 20 hidden units were used for most animals. And overfitting was not present using RBMs even when 100 hidden units were used. The proper size of the RBM would be dependent on the dimensionality of the input neural data. Given that the input data was from around 20 days/sessions for each animal and has many dimensions (45 for animal I10; 60 for A09, W09, L10 and J11; 75 for O10, T10 and B11; 90 for K11), 20 hidden units might be a relatively small number. This may be due to the rather simple binary classification task in this experiment, left versus right. And we would expect the size of the model to be subject to the complexity of the data and the decoding demand, and it may increase as larger populations of neurons are used to decode higher dimensional information.
Figure 20. Classification performance under different RBM size. The mean and standard deviation of SVM classification accuracy are plotted for each animal individually.

We then fixed the RBMs structure with 100 hidden units and compared SVM classification performance between using RBM generated features and using raw neural data. Results from all nine animals were shown in Figure 21. Classification accuracy is significantly higher (t-test, p<0.001) in seven animals when data has been processed by RBMs. For the other two animals (O10 and B11), RBMs did not have a strong effect on classification performance. One should also notice that the decoding accuracies of these two animals were also lower than the rest animals. One related factor would be that there were few direction modulated neurons in O10 and B11. Instead, most of their task related neurons in the cue onset period were modulated by the outcome (success or failure) of the
previous trial. The RBM failed to capture directional choice related patterns in their cases because there was little such information in the neural data at the first place. Taken together, these results support the expectation that RBMs are capable of extracting task related information which could benefit information decoding from neural firing activity.

Figure 21. Classification performance with versus without RBM processing. The mean and standard deviation of SVM classification accuracy are plotted for each animal.

5.3 Firing pattern adaptation with learning

In the previous analysis (Figure 21), training and testing data from all sessions of one rat were pooled together to train and test one model. It would be interesting to know how well this model fits single session’s data. If the model detected some common features in neural patterns from multiple days, we would expect to see stable decoding accuracies over sessions.
Results of using one model to decode single session’s data for each of the nine animals are presented in Figure 22. RBMs extracted features improved directional choice decoding in multiple sessions for each animal in most cases. The only exception is rat B11 which might be due to its lack of direction selective neurons so there was no consistent direction related firing patterns across sessions. The averaged single session decoding accuracy among all animals when using RBM modeling is 68.3%, which is significantly higher (t-test, p<0.005) than 65.0% averaged accuracy without RBMs.
These results further support the idea that certain features of the firing pattern presented in multiple days and could be captured using RBM models.

Since we recorded neural activity during animals learning a directional choice task, there could be systematic neural adaptations over sessions as animals’ task performance improves. Previous analyses in chapter 3 showed that firing patterns between left and right choice did become more separable as learning progressed. Those results were obtained by training individual SVM classifiers for each single session. Thus it is not straightforward to compare neural patterns over days in that circumstance since each day’s data was modeled by a different SVM classifier. In contrast, here we used one single model of SVM classification based on RBM features to interpret data from all recording sessions, so we could examine how this single model explains single session’s data, as an indicator of how the firing pattern represented by the model adapted over sessions.

In Figure 22, an increasing trend in the classification accuracy over sessions was observed in most animals. Rat A09 and I10 showed an opposite trend which could be due to their lack of understanding of the task: They did not complete the learning curve (below 70% behavioral accuracy except the last session of A09), while the other six rats (W09, L10, O10, T10, B11 and K11) all reached over 80% accuracy. To summarize results, sessions were divided into the three learning stages as defined before. And results from those six animals were used for analysis.
First, we reproduced previous results as shown in Figure 23A where SVM classifiers were built to model single session’s data. Classification accuracy increased over the three learning stages. Next, one single model (RBM feature extraction plus SVM classification) was built using all sessions’ data and then tested using data from single sessions. The classification accuracy using this model also increased over learning stages (Figure 23B), from a median of 60.4% in stage I to 70.5% in stage II and 69.4% in stage III (rank-sum test, p<0.05). Next, we calculated the difference in classification accuracy between the single-session model (Figure 23A) and the single model (Figure 23B). And Figure 23C summarized the statistics of this difference in single sessions, where a positive value indicated that the single-session model outperformed the other model. This measure increased from a median of 1.50% in stage I and 1.25% in stage II to 5.50% in stage III (rank-sum test, p<0.05).
It is not surprising that the decoding performance was better when the classifier was built to model individual session’s data. Intuitively, the single universal model was built not to cover detailed dynamics from individual days but to capture common features shared among all days. In this analysis, we intentionally chose the same number of data samples from single sessions to build the RBM model, so although the number of trials performed in each session was different, the model would not bias to certain sessions with more trials. In this sense, Figure 23B indicates enhanced firing patterns for choice direction discrimination developed over learning stages. Given the capability of RBM in capturing complex features, although a single RBM was built, it did not necessarily capture a certain single feature of the data. So the enhanced discriminability of neural activity by using this model may not suggest a better fit of data to one particular pattern later versus earlier in the learning process, but it’s likely that neural firing pattern adapted with learning to be more distinguishable between left/right choices and RBM model extracted those features along the course of adaptation.

The improved classification accuracy of using single-session SVM classifiers over using a single model of RBM and SVM based on all sessions (Figure 23C) may indicate unique features presented in individual sessions but did not get picked up by the single model, probably because they were not present in other sessions. The presence of these relatively transient features could come from several sources. First, they might be part of the neural dynamics induced by learning directional choice. While improved discriminability demonstrated by the universal model described the overall trend of the adaptation, there could be some day-to-day variations on top of that. Second, the same electrode may pick up neural action potentials from different neurons in different days,
especially in case of chronic recordings lasted for weeks. Besides, there might be variations in the recording environment from day to day including, for example, electronic noise from the recording system and the behavioral apparatus. So the presence of unique features in individual sessions should be normal.

These single day neural activity features did not considerably facilitate the decoding of directional choice until the third learning stage. In the first two learning stages, the single model decoded direction in single trials comparably well as those SVM classifiers trained and used to decode neural data session by session. These results may suggest that the neural adaptation described by the single model mainly occur during the first two stages but remained relatively stable in the last stage. Additionally only in the last stage, there was some flexibility over days which contributed to the neural codes of directional choice. Recall an animal’s learning curve in Figure 4. Notice that their behavioral accuracy improved quickly during the first learning stage but further improvements during the last two stages were slower. The above modeling results could then be interpreted collectively with those behavioral results that a systematic neural pattern adaptation occurred mainly during the early rapid learning stage, while when behavioral performance reaching a plateau neural pattern exhibited more variations from day to day. Another possible explanation could be that solving the directional choice task involved memory retrieval of the learned stimulus-response association in the third learning stage, while such rules were just being identified and gradually formed in the brain during the first two stages. And such difference may contribute to the unique neural firing patterns in the final learning stage as revealed by the RBM model.
5.4 Conclusion

Here in this chapter, spatiotemporal neural patterns corresponding to the animals making their directional choices were further explored using the RBM model recently advanced in machine learning community. This model was used to extract features in the neural firing patterns and to probe the neural adaptation during learning of the directional choice task. Results further supported that AGm and AGl neural ensembles adapted with learning although movements of executing directional choice were stereotypical. The RBM model revealed that neural patterns adapted to support enhanced discriminability between left side and right side choices mainly between the first two learning stages but remained relatively stable in the third learning stage. A comparison of choice direction decoding using RBM features built on all sessions’ data with using SVMs constructed for each individual session showed that choice direction related neural activity showed more variations from day to day in the third learning stage, which might be a reflection of neural adaptation related to the slow improving period later in the learning process. These findings added additional insight to the neural correlates underlying a learning process. Taken together, here we hypothesize that systematic neural adaptation and perhaps transition of neural states facilitate early rapid learning while later behavioral improvements involve additional adjustments around the new state.

The RBM training used an unsupervised learning procedure so features extracted from neural patterns were not restricted to those related to choice directions. Other task related factors, such as previous trial outcome, could also be explored using the RBM model. While RBM is a simple one hidden layer network, it could be stacked to build
deep networks. Building such artificial neural networks using real neural data will not only provide powerful data analysis tools, but may also bring new insights to our understanding of the real neural network.
CHAPTER 6 CONCLUSION

In this dissertation, the activity of rat AGm and AGl neural ensemble underlying the learning of directional choice was explored using analyses regarding mean firing rate, firing variability, and spatiotemporal firing pattern. The focus is how these neural characteristics adapt during learning the directional choice task. Significant changes were not observed in mean firing rate or trial-to-trial variability when measured over the three learning stages. On the contrary, spatiotemporal firing patterns exhibited improved discriminability between left side and right side choices. These results suggest a spatiotemporal neural coding scheme in a rat AGl and AGm neural ensemble that may be responsible for and contributing to learning the directional choice task.

These analyses used neural data when animals executed either a left or a right side movement. Stereotyped directional movements were observed and measured from video recordings of animal’s behavior. However, there might be unobserved changes of animal’s movements which could be a confound in neural activity analyses. Given this, although behaviors of freely moving animals are closer to their natural behaving state, restrained animals would provide more controlled models for neural study.

Throughout neural activity analyses in this dissertation, AGm and AGl neurons were not separated when neural ensembles were studied. This was restricted by the relatively small ensemble size we were able to simultaneously record consistently over a period of up to two months. The requirement of picking up units stably for such long period by single electrode further limited the number of units included in the study. Considering the functional differences between the AGm and AGl areas, it would be
interesting to see if there are differences in neural activity during the same directional choice learning process. Simultaneous recording of both AGm and AGl ensembles from the same hemisphere of rat’s brain would require elaborate configuration of electrode array placing and recording techniques.

Learning to choice a movement response according to certain sensory stimulus involves a neural network including the prefrontal cortex and the basal ganglia. The motor cortices studied here are connected with both areas and potentially take influence from them during the learning of the directional choice task. Monitoring neural activity in the prefrontal cortex and the basal ganglia during the same learning paradigm would bring a comprehensive understanding regarding how these areas together with the motor areas are related and involved in such learning process.
REFERENCES


APPENDIX A

AUTOMATIC MOVING TARGET TRACKING
AUTOMATIC MOVING TARGET TRACKING

In real-world scenarios, interactions among multiple moving targets can severely compromise the performance of the tracking system. Closely spaced targets are difficult to distinguish, and targets may be partially or totally invisible for uncontrolled durations when occluded by other objects. These situations are very likely to degrade the performance or cause the tracker to fail because the system may use invalid target observations to update the tracks. To address these issues, here we propose an integrated multi-target tracking system. A background-subtraction-based method is used to detect moving objects in video frames captured by a moving camera. The data association method evaluates the overlap rates between newly detected objects (observations) and already-tracked targets, and makes decisions pertaining to whether a target is interacting with other targets and whether it has a valid observation. According to the association results, distinct strategies are employed to update and manage the tracks of interacting versus well-isolated targets. This system has been tested with real-world airborne videos from the DARPA Video Verification of Identity (VIVID) program database, and demonstrated excellent track continuity in the presence of occlusions and multiple target interactions, very low false alarm rate, and real-time operation on an ordinary general-purpose computer. This system could potentially be used to monitor multiple animals’ movement trajectories and facilitate behavioral analysis in animal studies.
A.1 Introduction

Target detection and tracking is an essential problem in surveillance systems using radar, thermal, and electro-optical sensors. The objective is to obtain an observation of the target through the sensor at each scan and create a record of the target's trajectory by identifying and associating its observations over time. The target tracking problem has been studied intensely beginning with radar systems in the 1950s. Over the past two decades, significant attention has been paid to tracking targets in video sequences, which has been spawned by the proliferation of video sensors in a wide variety of applications such as military, remote sensing, robotics, and surveillance.

Some algorithms track targets automatically after the tracks have been initialized. For example, the mean shift tracking algorithm moves the track along the spatial gradient of certain feature similarity measurements to follow the target (Comaniciu et al., 2000). Particle-filter-based trackers (Arulampalam et al., 2002; Khan et al., 2004) iteratively generate particles to represent the probability density function of target states. However, these tracking algorithms themselves are not capable of initiating new tracks. A target detection module is always required to facilitate the automatic tracking of a variable number of targets. According to the application, some systems track certain types of targets, like vehicles (Zhai et al., 2007; Sun et al., 2006), pedestrians (Gavrila 2000), and faces (Lienhart and Maydt 2002) or heads (Benfold and Reid 2011), and usually use a feature-based target detector. In other systems, general moving targets are of interest, and they can be extracted using some form of foreground detection. For stationary cameras, a background model can be established from video frames under the assumption that the
background remains stationary. Foreground regions can then be detected by subtracting the background model from the current frame (Elgammal et al., 2000). A similar strategy also applies in moving-camera cases, where the current frame is usually subtracted from the aligned previous frame (Jung and Sukhatme 2004).

Detected target areas (observations) from consecutive frames must be linked together to form tracks. Data association algorithms are designed to solve this problem by associating observations of the same target from frame to frame. The multiple hypothesis tracking (MHT) (Reid 1979) approach exhaustively considers all possible associations where each observation could be associated with one of the existing tracks, initiates a new track, or represents clutter (a false alarm). A hypothesis is one feasible way of associating all observations up to the current frame, and all hypotheses are saved and evaluated. MHT is a complete algorithmic approach with the capability of initiating and terminating tracks, or in other words, can track a varying number of multiple targets. However, MHT's complexity grows exponentially, and the situation worsens when tracking a large number of targets for a long duration. Accordingly, several heuristic strategies have been developed to limit its growth. A comprehensive summary of MHT can be found in Blackman (2004). Unlike MHT, which makes absolute assignments, joint probabilistic data association (JPDA) (Fortmann and Scheffe 1983) updates a track using all observations, weighted by the probability that the observation originated from the track.

Classic multiple target tracking (MTT) algorithms (which include MHT and JPDA) assume one-to-one correspondences between observations and targets; that is, one
observation corresponds to at most one target and vice versa. This assumption may be applicable to radar tracking but is often violated in video tracking. As background subtraction algorithms are widely used in video tracking to detect foreground areas, one target might be detected as multiple "split" parts, and closely-spaced multiple interacting targets could be detected as one or more "merged" observations. In Perera et al. (2006), a track is terminated when the target is occluded or detected in merged observations. Afterwards, if the target is tracked again, the algorithm links the new track with the previously terminated one to maintain target identity. This track linking framework may work well for short-time occlusions and merges, but is not adequate for long-term or frequent target interactions. Genovesio and Olivo-Marin (2004) create virtual measurements by splitting real merged observations and merging real split observations. Kumar et al. (2006) organize feature (shape and color) matching results in dynamic programming tables to solve the data association problem with split and merge considered. These methods implicitly assume that a better observation of a target can be obtained by reorganizing observations generated by foreground detection. This assumption may hold in cases with a stationary camera, but may not apply when the camera undergoes substantial movement.

In moving-camera video, the background constantly changes due to camera motion, and no well-established background model is available as in stationary-camera cases. Moving targets are usually detected from temporal differences between consecutive frames. This may result in ill-detected target observations due to similar coloring across target surfaces so that interior target regions are not always detected as foreground. Methods have been developed to connect fragments within a convex hull
(Brown et al., 2006), assuming that targets have convex shapes. However, the situation degrades as multiple targets come close to and interact with each other, which results in non-ideal, dynamically varying, split and merged observations. A simple way to construct well-segmented observations by reforming the fragments would therefore be unlikely.

To prevent these imperfect observations from either disturbing the valid tracks or creating false alarms, an integrated tracking system is proposed herein with the capability of handling multiple interacting targets in video sequences with arbitrary camera motion. The main contributions of this paper are as follows: First, an overlap-rate-based data association algorithm is designed to provide information pertaining to splitting and merging, and to recognize target interactions; Secondly, a track maintenance and updating strategy is developed to accept and process observations from the foreground detections if they are deemed to be valid, or to employ alternate strategies (feature-based tracking) otherwise. Basic ideas of this tracking system had been reported previously (Mao et al., 2013). Here we present the system more completely and comprehensively regarding both the working principles and experimental results.

A.2 System Overview and Foreground Detection

A.3.1 System overview

The basic elements of the proposed system are consistent with those in a typical MTT system as shown in Figure A1. The foreground detection module extracts blobs (connected image pixels) as potential target observations from incoming video frames.
For each processed frame, the correspondences between new observations and existing tracks are built by the data association algorithm. By analyzing the data association results, situations such as splitting, merging, and more complex target interactions are recognized. This gives a hint as to whether or not the foreground blob is a good observation of the target. If it is, features of the blob are used to update the track; otherwise, the track will be searched for by matching its features profiled through previous frames. This strategy is effective in handling interacting targets which usually do not have well-segmented observations.

![Diagram](image)

**Figure A1.** A typical multiple target tracking system.

In real-world surveillance, target appearance can change substantially throughout the tracking duration due to illumination changes, target rotations, and changes in the camera's point of view. Therefore, the appearance model of a tracked target needs to be updated dynamically. During interactions, however, a target may be partially occluded and visually connected with other targets, in which case appearance model updates should be prevented. Failing to do so may result in loss of target track because targets are often tracked by matching their appearance models during interactions when valid
observations are not available, and inaccurate appearance models may distract the tracker in this context. This again emphasizes the necessity and importance of recognizing occurrences of target interactions. In addition to keeping a record of all existing tracks and maintaining a feature model for each track, the track management module also initializes new tracks and terminates inactive ones.

In modern target tracking systems, the problem is usually solved using a Bayesian filtering framework. Let $x_t$ be the target state at time $t$, which usually includes the target's current location in the image coordinates. $Z_t = \{z_0, z_1, \ldots, z_t\}$, where $z_t$ is the measurement at time $t$, and $Z_t$ includes all measurements up to time $t$. Then the target tracking problem can be stated as estimating $x_t$ given $Z_t$, or estimating the probability density function, $p(x_t | Z_t)$. This has already been well studied as recursive Bayesian estimation, and some applications like Kalman filtering and particle filtering have been widely used in various fields. In the proposed system, a Kalman filter is assigned to each tracked target to estimate its location from measurements, where a measurement could be either a blob from foreground detection that associated with the track, or a region found by feature matching, where the measurement used is dependent upon how the track was updated. A standard Kalman filtering algorithm is used for target location estimation and prediction, and a brief introduction of the filter can be found in Mao et al. (2011). Note that the Kalman filter performs three functions. First, it provides the optimal estimation (in the sense of linear dynamics with Gaussian noise) of target location after receiving a new measurement, thus smoothing the target's trajectory over time; second, it provides a prediction of a target's location, which is used in the data association module to find candidate observations; and third, when no measurement can be found (when a target is
totally occluded, for example) the predicted location is adopted and the track will be
updated accordingly.

Tracking results are presented to the user by drawing a bounding box
around the estimated target location. The color of the box together with a unique number
labelled alongside the box indicate target identity. A track file with these results can also
be generated. The performance of the tracking system can be evaluated by watching the
video with track labels and/or quantitatively when ground truth is available.

A.2.2 Foreground Detection

Moving target detection and segmentation is the first step in an automatic target tracking
system. This can be achieved by using temporal image differencing between the current
frame and the previous frame. Temporal image changes can originate from several
sources including camera motion, target motion, background clutter, and illumination
changes, among others. Since target motion is the only source of interest, effects of other
factors should be eliminated or suppressed.

Camera motion can be compensated for through image registration. This method
estimates a transform model to align two video frames of the same scene captured at
different times. Parameters of the transform model are estimated from a group of matched
feature pairs extracted from the two images. The previous frame is then resampled and
transformed into the same coordinates as the current frame. A survey of image
registration methods is available in Zitova and Flusser (2003). In the proposed system,
corner points (Shi and Tomasi 1994) are first extracted from the previous frame and then
tracked in the current frame using the Lucas-Kanade optical flow method. A global
projective transform matrix is then estimated from the motion vectors of all corner points. In practice, a distance constraint is employed to ensure that the chosen corner points are at least a certain number of pixels away from each other. This spreads out the corner points, which allows us to more accurately capture the global motion. It also guarantees that even if some of the points are from foreground moving targets, the number of these points will be limited so as not to unduly damage the image registration accuracy.

Heuristic criteria are also employed to check the validity of the transform matrix. This is necessary because bad matrices may arise due to featureless scenes, poor image quality, or unusual camera activities. Two principles are used to examine the transform: (1) the rectangular image should still be convex after transformation; and (2) the area of the image should not change dramatically after transformation given that the camera motion should be small between two consecutive frames assuming that the video frame rate is sufficient with respect to the apparent frame-to-frame velocity. In the case where the transform matrix is invalid, foreground detection will be bypassed and existing tracks will be updated using template matching.
Figure A2. Foreground detection. (a) current frame, (b) difference image, (c) binary image after thresholding, and (d) refined binary image.

Once a valid transform is performed, the difference image is obtained by subtracting the warped previous frame from the current frame. Background motion areas should be relatively stationary as compared with foreground motion areas, which would generate more noticeable differences. Pixel-level background-foreground classification is achieved by applying a threshold on the difference image, $D_t(x, y)$. A binary image, $B_t(x, y)$, is generated according to the classification rule in Equation A1:

$$B_t(x, y) = \begin{cases} 1, & \text{if} \frac{|D_t(x, y) - m_D|}{\sigma_D} \geq Th_1, \\ 0, & \text{otherwise}, \end{cases} \quad (A1)$$

where $m_D$ and $\sigma_D$ are the respective mean and standard deviation of the pixel values in the difference image, and $Th_1$ is the threshold. The binary image created by the
thresholding tends to be noisy and foreground pixels of the same target often form disconnected multiple pieces, as shown in Figure A2(c). Image processing techniques are applied to refine the binary image with the objective being to generate a single blob for each target area and eliminate false alarms arising from background areas. This process includes these steps: (1) remove areas smaller than a certain number of pixels to reject salt and pepper noise; (2) perform morphological closing operations to try and merge disconnected pieces; and (3) fill remaining holes within connected areas. Figure A2(d) shows a final foreground detection result with each blob (shown in black) forming one potential target observation.

Figure A3. Foreground detection under target interactions. EgTest02, frame 618 (a) original frame and (b) foreground areas, and frame 669 (c) original frame and (d) foreground areas. Targets are detected as split/merged observations, which are not stable but changing in different frames as targets move.
When targets are far apart, this foreground detection method will be able to provide well-segmented observations (Figure A2). As multiple targets frequently interact with each other, split and merged observations arise, and detected foreground areas are not stable but dynamically changing as targets move (Figure A3). This poses a challenge for the tracker to maintain continuous tracks and consistent target identity since valid observations are no longer available. Dedicated segmentation algorithms may help to produce better isolated observations, but these consume additional computational resources and are generally not suitable for real-time operation. Thus, in the current system, we accept these imperfect observations and attempt to compensate for them in the data association and track management modules.

A.3 Data association

The data association algorithm assigns new observations to corresponding tracks. Given the predicted location of a tracked target, its observation(s) should be geographically close to that prediction since natural targets tend to follow continuous-curvature trajectories. This principle has been widely used in data association algorithms as a gating procedure to preselect a set of potential observations for each track (Blackman 2004). Based on the same consideration, some system requires a good overlap between the prediction and observation in order to make an association (Song and Nevatia 2007). The data association method proposed here makes use of quantitative details of the overlap, so it is able to recognize situations like splitting, merging, and other complicated target interactions.
A.3.1 Overlap-rate-based data association

At time step $t$, when a video frame is captured, the algorithm sequentially examines how each new observation (a foreground blob) could be related to each existing track. Let $O_{t,i}$ denote the rectangular area surrounding the $i^{th}$ observation, and let $\hat{T}_{t,j}$ denote the rectangular area surrounding the prediction of the $j^{th}$ track. The overlap rates between the two are calculated as:

$$R_O = \frac{o_{t,i} \cap \hat{t}_{t,j}}{o_{t,i}} \quad \text{and} \quad R_T = \frac{o_{t,i} \cap \hat{t}_{t,j}}{\hat{t}_{t,j}}$$

(A2)

$R_O$ is the percentage of the overlap over the area of the observation, and $R_T$ is the measurement over the area of the track. When combined, the two rates together imply the area ratio between the observation and the track. Taking the area of the track as an estimation of the target size, a much smaller observation could only be a fragment of the target, and a larger one may contain some other area from the background or/and other targets. According to this, the type of association of each observation-track pair is defined in Table A1. Cases not included in the table, which are when either $R_o$ or $R_T$ are smaller than $Th_2$, are considered as “not associated”.

<table>
<thead>
<tr>
<th>$Th_2 &lt; R_T &lt; Th_3$</th>
<th>$R_T \geq Th_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Th_2 &lt; R_O &lt; Th_3$</td>
<td>Touch (T)</td>
</tr>
<tr>
<td>$R_O \geq Th_3$</td>
<td>Part (P)</td>
</tr>
</tbody>
</table>

Table A 1. Overlap-rate-based data association.
Figure A4. An illustration of overlap-rate-based data association. Types of association are labeled on edges. Some typical cases are: (a) an observation of a new target or clutter, (b) a merged observation, (c) split observations, (d) complicated interactions, and (e) an occluded target with no observation.

Each observation keeps a list of tracks, \( L_{O_{t,j}} \), that form an association with it, and each track also records its associated observations, \( L_{T_{t,j}} \). In the lists, the type of each association is saved as well. Figure A4 gives an example of the association results shown as a bipartite graph. These results are then analyzed to decide whether or not and how these foreground blobs are used to update the tracks.

A.3.2 Non-overlap association

It is conceivable that the track does not closely follow the target and thus cannot be associated with the target's observation by overlap-based methods. This situation is mostly caused by occlusions or target interactions when the track updating relies on feature matching or prediction. Typically, the track may drift away from a target under long-term occlusion due to accumulated prediction error, and feature matching could be compromised by appearance changes, partial occlusion, another similar target, or even
background regions. In these cases, when the target comes out of occlusion or interaction and a valid observation is available again, it will be treated as a new target, and thus ruin the track continuity and target identification if the observation could not be associated with its existing track. Therefore, an additional non-overlap association step is developed to address this problem.

Following the overlap-rate data association, a second round of association is carried out between \( \{O_{t,i}|L_{O_{t,i}} = \emptyset\}\) and \( \{T_{t,j}|L_{T_{t,j}} = \emptyset\} \), and only one-to-one correspondences are considered. The association criteria are: (1) the distance between the observation and the track should be within \( Th_D \) pixels; and (2) the observation should be along the track’s motion direction, i.e., \( M_{\text{ang}} \in [0, Th_A] \cup [180^\circ - Th_A, 180^\circ] \), where \( M_{\text{ang}} \) is calculated as

\[
M_{\text{ang}} = \text{angle} (\overrightarrow{T_{t-1,j}O_{t,i}}, \overrightarrow{T_{t-1,j}T_{t,j}}),
\]  
(A3)

and (3) the rectangular area that the observation and the track occupy should be similar in size so that the size measurement, \( M_{\text{size}} \) in Equation A4 should be greater than a threshold, \( Th_S \), where \( w_O \) and \( h_O \) are the respective width and height of the observation region and \( w_T \) and \( h_T \) are those of the track.

\[
M_{\text{size}} = \exp \left[ -\left( \frac{(w_O - w_T)^2}{2\sigma_w^2} + \frac{(h_O - h_T)^2}{2\sigma_h^2} \right) \right].
\]  
(A4)

These criteria are chosen under the assumption that targets maintain a consistent motion pattern and appearance after occlusion or interaction. Other criteria and target features may be used according to the application.
When an association is made in this step, the observation and track are added to each other’s association list just as in overlap-rate-based association, and the type of association is labeled as “N”, meaning “non-overlap”. There is usually at most one observation that qualifies for this non-overlap association for one track, given that most other observations and tracks have already been handled by the overlap-rate-based association. In cases where there are multiple candidates, the aforementioned measurements could be combined to decide a best match.

A.3.3 Association result synthesis

The association results are recorded in the association list of each observation and track, including the indices of the tracks or observations it associated with and the type of each association. These results are analyzed and synthesized to decide two things: First, whether a track has a valid observation from foreground detection; and second, whether a target is involved in any interaction with any other target.

*Interacting targets labeling.* When a target is interacting with other targets, it is unlikely that the current foreground detection method can provide a well isolated observation for that target, so the proposed system does not use the observation to update and interacting target, and interacting targets are processed differently in the track updating module, which will be introduced later. An existing track is recognized as being in an interaction in one frame if for $T_{t,i}$, there exists $O_{t,i} \in L_{T_{t,j}}$ so that $sizeof(L_{O_{t,i}}) > 1$.

In other words, when an observation associates with more than one track, then all tracks associated with the observation will be labeled as “interacting” (Figure A4(b)(d)).
Given the relatively high frame rate (e.g., ~30 fps) of surveillance cameras compared with the speed of common real-world moving targets, a multiple target interaction, e.g., one car overtaking another, may last for several successive frames. Rather than labeling whether or not (as a boolean variable) a target interacts with other targets in each frame, a nonnegative integer, $n_{\text{interact}}$, is assigned to each track as the interacting index. Whenever a target is recognized as interacting, its interacting index is assigned to a fixed number, $N_{\text{interact}}$. This number decreases by one if interaction is not detected in one frame, until it drops to zero. A track is treated as an interacting target in the subsequent track updating procedure when its $n_{\text{interact}}$ is greater than zero. This index number is introduced to briefly extend the duration of a recognized interaction. This is necessary because there is still a chance that the algorithm would fail to recognize interacting targets in a certain frame when their merged image area was detected as several irregular split foreground areas, and they made one-to-one associations with targets. In this case, the targets will not be recognized as interacting in that frame, but their interaction history detected in the immediately preceding frames alerts the system that they may still be involved in interactions. The interacting index number thus makes interaction detection more robust and reliable.

*Association confirmation.* While handling occlusions and interactions is critical, tracking widely-spaced targets is a fundamental objective in any tracking system. These targets are highly likely to have well-segmented observations associate with them and their tracks will be updated using these observations once the associations are confirmed.
If an association is one-to-one, i.e., an observation and a track are the only elements in each other's association list, the association type is checked to confirm the assignment. If the association type is “M” or “N”, the association is confirmed immediately and the observation is assigned to the track. In the case of “P” or “L”, the size of the observation is evaluated using Eq. A4. A reasonable observation should be similar in size to the target, otherwise it (either a small fragmented piece or a larger area containing background areas) will be ignored. To approve a “T”-type association, since the confidence level is lower due to poor overlap, in addition to verifying the size of the observation, \( n_{interact} \) of the track is required to be zero. This is because in an interaction, a nearby observation is likely to be from another target, so we conservatively deny the association even though it is one-to-one. The criteria used here are convenient to compute and shown to be effective with the test data. Other standards may apply and could be developed as needed.

Some cases when multiple observations associate with one track are processed as well. When two observations both associate with one single track as type “P”, they are combined as one whole observation (a rectangular area covers both observations) and assigned to the track (Figure A4(c)). When one “M” or “L”-type together with another “P” or “T”-type observation associate with one track, the “M” or “L”-type observation is assigned and the other observation is discarded. Only these two types of multiple-to-one associations are handled in the current system, with the reason being mostly heuristic.

Only in the aforementioned cases is an observation assigned to and used to update a track. These cases may be a small portion of all possible association scenarios, but they
cover the most common cases for tracking widely-spaced targets. A track without any confirmed association only indicates that the foreground detection method failed to find an valid observation. Other methods like feature-based tracking may still be able to maintain the track, however. In the following section, tracks with and without assignments, including some involved in interactions, are updated using different strategies. Motion-based and feature-based tracking methods are combined to maintain continuous target identification in challenging scenarios.

A.4 Track management

This track management module controls the entire life cycle of all tracks, including their initialization, updating, and termination. These issues are closely related to the overall tracking performance. The objective of the proposed system is to perform automatic track initialization and termination with low false alarm rate, and continuous target tracking through occlusions and multiple target interactions.

A.4.1 Track initialization and termination

The foreground detection method introduced in Sec. A.2 automatically extracts observations of moving targets from video frames. Track initialization corresponds to finding those observations that have been detected consistently and begin to record their trajectories. When foreground blobs are first detected in a frame, each one of them is used to initiate a new potential track. Blobs detected in subsequent frames are checked to see if they associate with (by nearest-neighborhood data association) any potential track.
Only one-to-one association is considered and the potential track updates its location when an association is found. Any potential track that has been successfully updated for a certain number of frames will be confirmed as a true target (thus initiating a formal track) if it passes certain motion pattern tests (Li et al., 2010), which require a smooth trajectory so as to reject false alarms. Observations extracted in later frames are first compared against existing tracks for possible associations as described in Sec. A.3. Only those that do not associate with any track in any form are used to associate with potential tracks. If associations still cannot be made, those observations spawn new potential tracks. This procedure is carried out for every frame so as to automatically initiate new tracks as new targets enter the camera's field of view. It also effectively suppresses false alarms without losing sensitivity to true moving targets. Note that only confirmed tracks are output to the user.

A track is terminated only when it moves out of the boundary of the frame. If a previously-tracked target re-enter the frame, it will be tracked as a new target, and continuous target identification will not be preserved.

A.4.2 Track deactivation and reactivation

As will be discussed later, a track will be assigned a predicted location if it fails to get updated by other methods. If this occurs for a certain number of frames, the track will be switched to an inactive mode whereby it is still updated but not presented to the user (i.e., it is invisible). This can happen when a track can no longer follow the target for whatever reason, and would otherwise look like a false alarm. When a target is occluded, for example, there is neither associated foreground blob nor any feature-matching
information. The track is not immediately terminated but only turns invisible to accommodate the possibility of resuming the track when the target is visible once again. This deactivation strategy may also affect the tracks of interacting targets, which do not have valid foreground observations and for which feature matching may fail because the targets may be partially occluded and/or visually connected with other targets. Note that interacting targets may still be visible (even partially) and the predictions may still follow the targets. Therefore, an exception to the track deactivation strategy is that it does not apply to interacting targets.

Some of the deactivated tracks may be reactivated when their corresponding targets emerge from an occluded state. Inactive tracks do not participate in overlap-rate-based data association since they may travel in any direction along their predicted paths and may disturb the correspondences between targets and their observations. Reactivation can be made only upon non-overlap associations.

A.5.3 Track Updating

Our previous tracker (Mao et al., 2010), which utilized only motion detection and filtering, was able to track widely-spaced targets and handle simple interactions (e.g., a vehicle overtaking another). In order to track targets through more complicated interactions when observations from motion detection are usually unavailable, a track updating mechanism has been developed and is shown in Figure A5.
For a track that has an observation confirmed and assigned by the data association module, the update mostly depends on this observation. A similarity measure between the track and the observation is calculated in the same way as DST in Li et al. (2010), but with target features that include the mean value of the three color image channels, width and height, and velocity being used instead. If the similarity measure is greater than a threshold, $Th_4$, the location of the track (the bounding box) is updated according to the observation's location. If the similarity is greater than a higher threshold, $Th_5$, then the appearance model of the track is updated, including a subimage of the target in the current frame in addition to the features mentioned above. If the similarity does not reach $Th_4$, the track will not be updated using the observation, but treated the same as other tracks that do not have an assigned observation.

Figure A5. Overview of track updating.
For any track that is not updated by an observation, a small area within the frame is searched to find a best match. A template matching method is used in the current system, using the target’s subimage as the template. When a track is recognized as interacting, the search area is along the track's motion direction; otherwise, it will be searched for in a rectangular neighborhood around the predicted location. Again, $Th_4$ and $Th_5$ are the thresholds used to measure the correlation coefficient at the best matching location, and to decide whether to use the best matching area to update the track's bounding box and appearance model, with the exception that the appearance model is not allowed to be updated if the track is deemed to be interacting. If the template matching method cannot find a match for the track, the predicted location from the filtering module will be used to update the track.

As mentioned in Sec. A.2, it is important to keep the target's appearance model updated because it may undergo substantial changes, especially when tracking in an outdoor, unconstrained environment for a long duration. The subimages of the targets get updated by simply replacing the old ones. Features (color values, sizes, and velocities), however, are updated according to Equation A5:

$$x^t = (1 - \alpha)x^{t-1} + \alpha x_t,$$

(A5)

where $x^t$ and $x^{t-1}$ are respectively the current and previous feature value after updating, $x_t$ is the value just observed, and $\alpha$ is the update strength. Equation A5 lends robustness to the appearance model robust in the presence of noisy observations.
This track updating framework uses foreground detection (motion information) as a basic source of target observations. When motion detection cannot provide well-extracted observations, the system tries to locate the target using feature-based matching methods. If both attempts fail to update the track, it will use the predicted location. Following this track updating logic, different types of information, including frame differences, target appearance features, and target motion patterns, are prioritized and combined to increase the chance of tracking targets continuously through challenging scenarios such as multiple target interactions and occlusions.

A.5 Experimental results

The proposed moving target detection and tracking system has been tested using real-world video sequences. The test sequences were taken from the DARPA Video Verification of Identity (VIVID) dataset (available upon request at https://www.sdms.afrl.af.mil), and include the five "EgTest" sequences ("EgTest01" to "EgTest05", http://vision.cse.psu.edu/data/vividEval/main.html) and the two sequences "V3V10003\_004" and "V4V100014\_060" ("video06" and "video07," respectively). Test sequences were composed of color imagery taken from a single camera mounted on an aerial vehicle, and all of the targets in the sequences were ground vehicles. The original image resolution was $640 \times 480$ pixels and the frame rate was 30 frames per second (fps). The VIVID dataset was constructed for the purpose of video target tracking evaluation and provides a wide variety of troublesome scenarios including arbitrary and abrupt camera motion, occlusions, multiple target interactions, fast-moving targets, and
out-of-focus video, among others, which makes it an excellent dataset for testing visual trackers.

The proposed system was implemented in C++ with OpenCV 1.1pre1a. To test the system in real time, the video sequences were spatially downsampled to $320 \times 240$ pixels and temporally downsampled to 10 fps. The parameters used for system evaluation were $Th_1 = 2.5$, $Th_2 = 0.05$, $Th_3 = 0.5$, $Th_4 = 32$, $Th_5 = 30^\circ$, $Th_6 = 0.9$, $Th_4 = 0.85$, $Th_5 = 0.9$, and $\alpha = 0.4$.

Table A2. Tracking performance evaluation.

<table>
<thead>
<tr>
<th></th>
<th>EgTest01</th>
<th>EgTest02</th>
<th>EgTest03</th>
<th>EgTest04</th>
<th>EgTest05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match (%)</td>
<td>95.00</td>
<td>93.02</td>
<td>85.88</td>
<td>60.00</td>
<td>88.89</td>
</tr>
<tr>
<td>Size ratio</td>
<td>1.00</td>
<td>1.23</td>
<td>0.78</td>
<td>1.19</td>
<td>0.88</td>
</tr>
<tr>
<td>Time (ms)</td>
<td>58.1</td>
<td>81.6</td>
<td>96.4</td>
<td>69.6</td>
<td>53.1</td>
</tr>
</tbody>
</table>

Several quantitative results are presented in Table A2. Ground truth was available for one target in each of the five EgTest sequences for every 10th frame, as provided in Ref. 26. A match is defined when the bounding box of the track captures the target center of the ground truth. The percentages of frames where the tracking result matches the ground truth are listed. High percentages are achieved except in EgTest04, where the target keeps entering the boundary areas of the frame, which is not covered by image differencing. In this case, the algorithm often tracks the long shadow behind the target, which results in a low matching percentage because the ground truth only covers the target body. All moving targets in EgTest04 are detected and tracked with good track.
continuity, though the main body of the target may not always be tracked with the shadow sometimes being tracked instead in full or in part. The size ratio in Table A2 is calculated as the area of the track (in pixels) divided by the ground truth area, and averaged over all available frames. This gives a basic measure of the target segmentation performance. The results are acceptable given the simple thresholding-based foreground detection method. Finally, average processing times for each frame are below 100 ms (on a PC with an Intel Core 2 Duo CPU running at 2.83 GHz with 4 GB of RAM), which enables us to process the video in real time at 10 fps. Finally, in over 10,000 frames of video over all sequences, only one false alarm was observed.

Tracking results were also visually inspected since the ground truth was rather sparse and the quantitative analysis often lacks specific details of certain scenes of special interest. Figure A6 demonstrates continuous tracking and target identification in spite of substantial target appearance changes. A target experiencing occlusion is shown in Figure A7. The track (target 2) took on predicted locations while the target was occluded and invisible, and the target was tracked with no loss of continuity after emerging from occlusion.
Figure A6. Targets undergo substantial appearance changes. (a-c) Target 1, 2 and 3 in EgTest02, frame 66, 336, and 1299. (d-f) Target 2 in EgTest05, frame 468, 996, and 1422.

Figure A7. Tracking through complete occlusion. (a-d) Target 2 in EgTest04, frames 648, 666, 717, and 744.
Scenes containing target interactions are illustrated in Figure A8. In the first example, Figure A8(a-c), target 1 overtakes target 9. In Figure A8(d-f), target 6 and target 10 pass each other while heading in opposite directions. A more complicated and challenging case is shown in Figure A8(g-i), where two sets of three vehicles each pass each other very closely and at very low velocities. Some of the targets are partially occluded during the interactions, and target areas are visually connected, making them difficult to separate. Also, the interactions among the targets also change dynamically as the targets move. In spite of these very challenging conditions, the proposed system successfully tracked all six targets and maintained correct target identification.
Figure A8. Target interactions. (a-c) EgTest01 frame 1638, 1674, and 1713. One vehicle (target 1) overtakes another (target 9). (d-f) Video07 frame 1200, 1230, and 1272. Two vehicles (target 6 and 10) pass each other. (g-i) EgTest02 frame 408, 558, and 768. Two sets of three targets (targets 1, 2, and 3, and targets 4, 5, and 6) pass each other.

Several track trajectories are presented in Figure A9. Three targets in EgTest02 (Figure A9(a)) always remained within the camera's field of view and were tracked for the entire duration of the sequence. In Figure A9(b), two sets of vehicles had just passed each other. Figure A9(c) shows a scene where a group of vehicles take a rapid turn. And in Figure A9(d), the target was tracked through occlusions by the trees.
A.6 Conclusions

In this work, an automatic, multiple target detection and tracking system has been proposed, with the ability to handle multiple interacting targets viewed from a moving platform. The novel overlap-rate-based data association algorithm provides the ability to recognize split and merged observations, and in turn, multiple target interactions. The track update module uses motion detection results, target appearance features, and target motion patterns to track targets in different contexts. Together with other modules, the system provides an integrated solution to the multiple-target-tracking problem. When tested using real-world aerial video sequences, the system demonstrates excellent performance, including automatic track initialization and termination, near-zero false
alarm rate, robustness to target appearance changes, and continuous tracking through occlusions and multiple target interactions. Finally, the computational tractability makes the system ideally suited for real-time video target tracking applications on both general-purpose computing platforms and embedded systems (Gao et al., 2013).

References


