Cortical Inference of Visual Input based on Retinal Spikes

Esha Uboweja

CMU-CS-15-128

August 2015

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Prof. Tai Sing Lee
Prof. Steve Chase

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

Keywords: eye motion, retina, cortex, object motion, object reconstruction, mental images

"In general, we are least aware of what our minds do best."

- Marvin Minsky

To everyone who taught me this:

If you seek, you may never find.
If you try, you may only guess.
If you beg, you will never get.
If you hope, you may only dream.
If you turn, you will surely lose.
But,
If you burn, you will deserve to desire.

- Esha Uboweja, 2014

Abstract

Vision is one of the most vital and complex functions of the brain. Photoreceptors in the eye encode light intensities and ganglion cells in the retina generate spike trains to convey information to the visual cortex. These spike trains are sparse and stochastic and a mental image of the visual scene can be recovered by the brain only by integrating the spikes over time. However, the left and the right eyes jitter incessantly and independently every millisecond, preventing a simple integration process. The brain has to infer the image representation of the visual scene while simultaneously computing the net eye jitter trajectory. This is a difficult chicken-and-egg problem. It is intriguing that the visual system can use information from sparse and corrupted spike trains to infer what the eyes see, clearly with every intricate detail. Burak et.al. demonstrate the inference of binary images of static scenes in the presence of eye jitter using retinal spike train input via a Factorized Bayesian Decoder. But how does the visual cortex infer details of objects moving in a scene? Since objects move continuously in space over time, the number of spikes emitted by the retina from any one object location are even more sparse.

We seek to generalize this Bayesian factorization framework to deal with (1) a dynamic scene with an object moving relatively to a background, (2) gray-level and more naturalistic images. The system seeks to reconstruct and infer details of the moving object, and decompose that from the recovered image of the background. We demonstrate the feasibility of this framework in solving the puzzle of how the brain can construct a stabilized image of dynamic visual scenes in the presence of incessant eye movements.

Acknowledgments

I would like to thank my advisor Prof. Tai Sing Lee for encouraging me to explore many ideas and for useful redirection and advice. I would also like to thank Prof. Steve Chase for being a part of my thesis committee and reviewing my work.

Contents

Chapter 1

Introduction

Our brain is able to encode natural scenes and moving objects in spike trains emanating from ganglion cell neurons in the eye's retina. These spike trains are sparse and noisy. Further, our eyes move randomly so that even stationary images jitter on the retina. The visual system should infer a stabilized mental image from these spike trains so that we can see stationary and moving objects clearly.

Stabilization of the mental image is important for two reasons:

1. So that we may see the world as it is: If we don't perform stabilization, stationary objects would appear as moving objects, and it would be hard to track the actual motion of an object, such as when we are trying to catch a ball.
2. For stereo-vision: We know that by using 2-D images from both left and right eyes we can match corresponding points of objects present in both images and compute disparity, which helps in depth computation. These images must be stabilized so that we may correctly match corresponding object points and correctly compute disparity.

One may argue that if it is very important to know eye jitter trajectory, then it seems necessary for the brain to have appropriate circuitry to communicate this information from the retina to the cortex for each eye. However, eye jitter is the combined result of eye, head and body movements, and even with the presence of a dedicated vestibulo-ocular reflex pathway to reduce it, a significant amount of global image jitter needs to be computed by the cortex [5].

A question that arises given such constraints and assumptions - how do we see objects in motion? Since spike trains transmitted via optic nerves to the cortex are already sparse, the spikes that result from neurons viewing an object shifting rapidly to different positions will be even fewer in number. Reconstructing a moving object from such spike trains is an intriguing problem, and this thesis examines how this problem can be solved in certain kinds of visual scenes.

We analyze a Bayesian framework proposed by Burak et.al., namely Factorized Bayesian Decoder (FBD) [4]. Similar to the Expectation-Maximization technique in machine learning, FBD computes eye jitter $x(t)$ in 2-D, and inferred 2-D image of the scene concurrently, using simulated retinal spike input for a static scene. Note that while there are many different kinds of cells in the retina [5], each performing its specific function, we use only a simulation of retinal ganglion cell spike trains conveying image pixel intensities. The aim of this thesis is to present and discuss a computational framework in the primary visual cortex and not present a simulation of what happens in different retinal cells biologically.

1.1 Thesis Overview

The thesis is organized as follows. Chapter 2 presents the *Factorized Bayesian Decoder* (FBD) by Burak et.al. for decoding spike trains generated by viewing binary images of static objects. Chapter 3 presents an extension of FBD for decoding spike trains generated by grayscale images of static natural scenes. Chapter 4 explores the problem of reconstructing stable images of moving objects from spike trains generated by viewing binary images of dynamic scenes. Chapter 5 analyzes the same problem using spike trains generated by viewing moving objects in natural scenes (grayscale). The thesis concludes with a discussion in Chapter 6.

Chapter 2

Factorized Bayesian Decoder

In 2010, Burak et.al. proposed a Bayesian model for inferring stabilized images of static objects and net eye jitter, namely the Factorized Bayesian Decoder (FBD) in the visual cortex [4]. This chapter explains how FBD works when the eye views binary images of static objects. The next chapter discusses how FBD can be used to view static objects in natural scenes (grayscale).

2.1 Setup

To simulate how the visual cortex may decode spike trains, we need a model for each of the following:

1. The visual stimulus presented to the system.
2. The eye jitter - how the visual stimulus gets displaced with respect to (w.r.t.) the retinal ganglion cells.
3. The spike trains generated by ganglion cell neurons in the retina in response to the visual stimulus.

In this chapter, we consider visual stimuli that are black and white images of static objects. The input image is represented as a function $\{s_i\}\,\forall i$, where i is the index of pixel location in the image. We are considering 2-D images, so that a 2-D image of size $m \times n$ has a unique index i for every pixel.

Burak et. al. model the retina as a grid of ganglion cells of similar type in a small patch of the fovea, so that each ganglion cell corresponds to one pixel in the stimulus. So a grid of size $m \times n$ will observe an image of the same size. Each neuron fires spikes at a firing rate proportional to the value of the pixel observed by it, as described in the following equation:

$$\lambda(v) = \lambda_0 + \Delta\lambda v \tag{2.1}$$

where λ_0 is the firing rate of the neuron when it sees a black (OFF) pixel, λ_1 is the firing rate of the neuron when it sees a white (ON) pixel, $\Delta\lambda = \lambda_1 - \lambda_0$ and v is the value of the concerned pixel. This equation requires that $\lambda_1 > \lambda_0$ for spike count to be unambiguous when encoding the brightness of the pixel.

FBD models each neuron so that $\lambda_0 = 10$ Hz and $\lambda_1 = 100$ Hz. So, the neuron fires more spikes when it observes an ON pixel. Thus, a spike from such a neuron at any timestep t indicates that the neuron observed something (as the spike may be from an ON or OFF pixel), but a high count of spikes from the same neuron over time increases the probability that the neuron observed a white pixel. Also, $\lambda_1 = 100$ Hz means that over 1 second or 1000 milliseconds (ms), the neuron fires spikes when observing a white pixel continuously only 10% of the time. FBD solves the problem of seeing objects clearly when we fixate on it. Human fixation time is approximately 300 ms. Therefore, a neuron observing ON pixels will fire about 30 spikes on average during fixation, while a neuron observing OFF pixels will fire about 3 spikes on average during fixation. The neuron doesn't emit spikes for an ON pixel all the time. This lack of spikes is the sparsity which denotes absence or loss of information about pixel intensity.

A neuron k generates a spike train $\{g_k(t)\}\,\forall t$ for the duration of fixation.

Our eyes move randomly, so Burak et.al. model net eye jitter as a random walk. When we consider a patch of the fovea and the stimulus falling on it, eye jitter would displace the foveal patch so that only part of the stimulus is visible to that patch. This is mathematically represented as displacement x of the eye w.r.t. the stimulus being observed.

The eye jitter is thus modeled as a set of displacements $\{x(t)\}\,\forall t$. Since the entire eye gets displaced w.r.t. the entire image, for any neuron k and image pixel i, we can state that $x = k - i$. See Figure 2.1 for how the jitter direction works. The figure shows an image viewed by a red retinal grid patch of neurons. A positive displacement will shift the retina towards the right/down and a negative displacement will shift the retina towards the left/up (depending on direction of displacement).

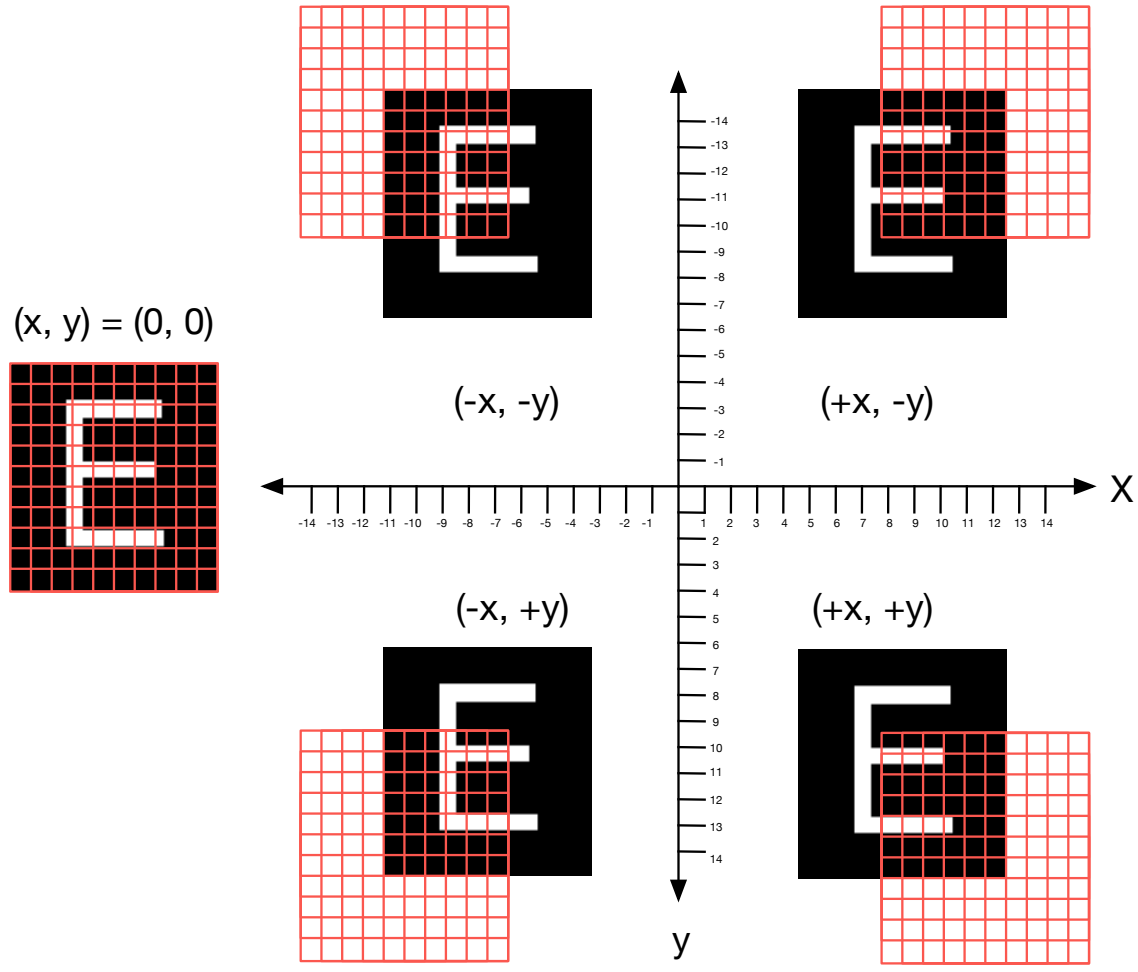


Figure 2.1: Model of the retinal displacement jitter (x, y) in 2-D

So, a neuron k fires a spike at timestep t ($g_k(t) = 1$) when it observes that a pixel i is ON ($s_i = 1$). Neuron k is at displacement x w.r.t. i so that $x = k - i$. When the displacement x is 0,

$k = i$ and the image stimulus aligns with the neuron grid.

2.2 Building a decoder using Bayesian dynamics

To infer a stabilized image we need to estimate eye jitter simultaneously, and we can do so by keeping track of the joint probability of every possible binary image $\{s_i\}\,\forall i$ for every possible displacement x. That is, we need to track $P(s, x, t)$ at every timestep t, where $P(E)$ denotes the probability of an event E. For a binary image stimulus of size $m \times n$, there are $2^{m \times n}$ possible images as each pixel $s_i \in \{0, 1\}$. Since there are so many possible images, it's difficult and inefficient to track $P(s, x, t)\ \forall s, x, t$. To solve this problem Burak et.al. proposed a factorized approximation to the joint distribution $P(s, x, t)$ using Bayes rule.

Factorized Bayesian Decoder (FBD) calculates $P(s, x, t)$ as a product of independent probability distributions: $P(x, t)$, the probability of displacement x at timestep t, $P_i(s_i, t)$, the probability of pixel value s_i at pixel location i at timestep t ($P_i(E)$ denotes the probability of event E at pixel location i). The approximation is stated as:

$$P(s, x, t) = P(x, t) \prod_{i=1}^{N} P_i(s_i, t) \tag{2.2}$$

where N is the total number of pixels in the image stimulus. This factorized approximation ignores any correlation between pixel values s_i, s_j or jitter displacement x and pixel value s_i, for any i, j. A derivation of this decoder is presented in the appendix section of [4] and employs Bayes rule for this factorization.

At every pixel i, we keep track of $P_i(s_i, t)\,\forall s_i$. For binary images, $s_i \in \{0, 1\}$. By law of total probability, $P_i(s_i = 0, t) + P_i(s_i = 1, t) = 1$, hence we can track only $P_i(s_i = 1, t)$ and calculate $P_i(s_i = 0, t) = 1 - P_i(s_i = 1, t)$ as needed. Denote $m_i(t)$ as $P_i(s_i = 1, t)$.

Further, we need to know how many possible displacements x the eye can jitter. The displacement x is in 2-D. Our net eye movement is bound by a maximum possible jitter because how far we can move our eyes due to head movements or eye movements is limited. For our implementation of FBD, we assume that the eye can jitter only about 20 pixels in positive or

negative direction of x or y axes. At $x = (0, 0)$, the image pixels s_i would align one-to-one with the grid of ganglion cell neurons k, so that the entire image stimulus is visible on this ganglion cell grid patch.

Given ganglion cell spike trains $g_k(t)\,\forall t$ for all neurons k, we want to infer a stabilized image $b_i(t)\,\forall i$ and eye jitter $x(t)$ at every timestep t.

Suppose at timestep t we don't see any spikes. Then, our belief in probabilities of displacements x should weaken, as we have no support information. So, our belief $P(x, t)$ will diffuse as:

$$\frac{dP(x,t)}{dt} = D\nabla^2 P(x,t) \tag{2.3}$$

Here, D is the diffusion coefficient used to generate the random walk trajectory for jitter $x(t)$ and ∇^2 denotes the Laplacian operator. Hence $P(x, t)$ gets updated as:

$$P(x, t_+) = P(x, t_-) + D\nabla^2 P(x, t_-) \tag{2.4}$$

Here, t_- is the time before update, and t_+ is the time after update.

Further, the absence of spikes should weaken our belief in the probabilities of pixels being ON, as a low count of spikes over time decreases the probability of neuron(s) observing ON pixels. So,

$$\frac{dm_i(t)}{dt} = -\Delta\lambda(1 - m_i(t))m_i(t) \tag{2.5}$$

Hence, $m_i(t)$ gets updated as:

$$m_i(t_+) = m_i(t_-) - \Delta\lambda(1 - m_i(t_-))m_i(t_-) \tag{2.6}$$

The function $(1 - m_i(t))m_i(t)$ is highest when $m_i(t) = 0.5$. $m_i(t) = 0.5$ implies that pixel i has equal probability of being ON or OFF. So the absence of spikes will make $m_i(t) < 0.5$, which further implies that pixel i is probably an OFF pixel. Also, $(1 - m_i(t))m_i(t)$ is 0 when $m_i(t) = 0$ or $m_i(t) = 1$, that is, when we are certain that a pixel is OFF or ON, we don't update our belief and this seems appropriate. A plot of the function $(1 - m_i(t))m_i(t)$ is shown in figure 2.2

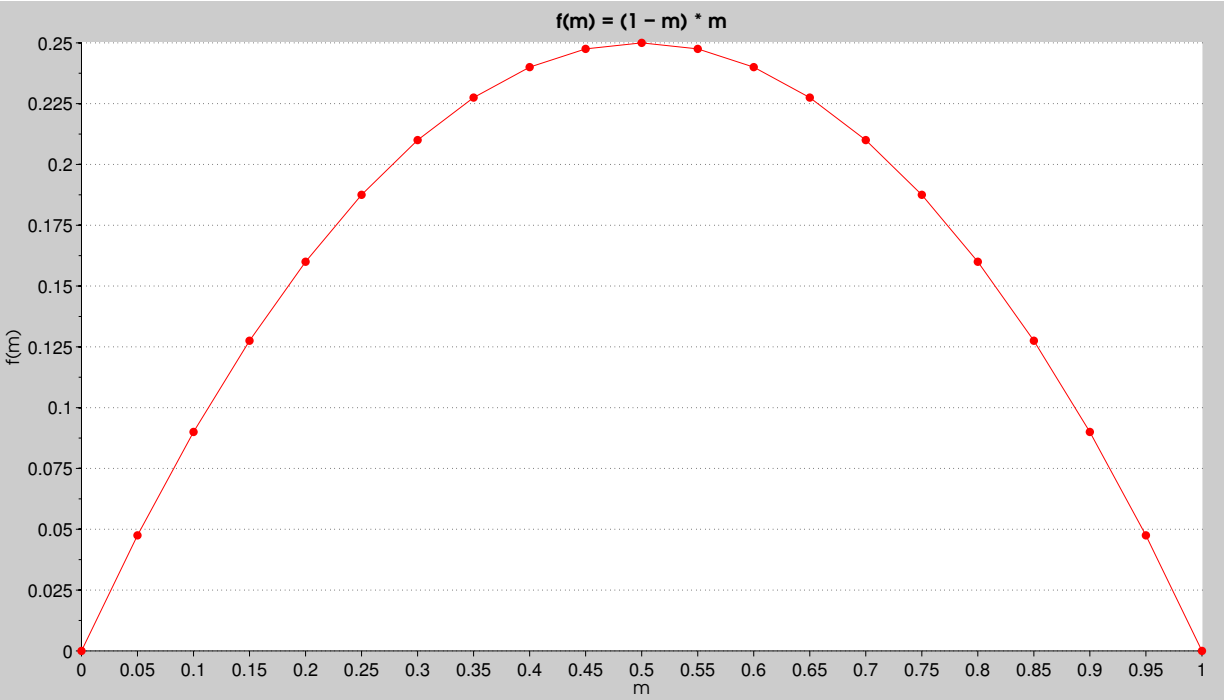


Figure 2.2: Plot of the function $f(m) = (1 - m) * m$

Suppose that at timestep t, we observe a spike from ganglion cell neuron k ($g_k(t) = 1$). This means, we can use the spike at k to update our belief $P(x, t)$ for all x as follows:

$$P(x, t_{k+}) = P(x, t_{k-}) \frac{(\lambda_0 + \Delta\lambda m_{k-x}(t))}{R_k(t_{k-})} \tag{2.7}$$

where

$$R_k(t) = \lambda_0 + \Delta\lambda \sum_x m_{k-x}(t) P(x, t) \tag{2.8}$$

where t_{k-} indicates the time before updating a quantity due to a spike from neuron k, t_{k+} indicates the time after updating a quantity due to a spike from neuron k.

$(\lambda_0 + \Delta\lambda m_{k-x}(t))$ gives us the expected firing rate at pixel location $(k - x)$. If $m_{k-x}(t)$ is high, then $(\lambda_0 + \Delta\lambda m_{k-x}(t))$ is high, and the probability of displacement x is high. However, if $m_{k-x}(t)$ is low, then x is not a likely displacement because neuron k fired. $R_k(t)$ is the expected firing rate over all displacements x.

$R_k(t)$ is the expected firing rate due to spike of neuron k at all possible displacements x.

By definition, if $R_k(t)$ is high and $(\lambda_0 + \Delta m_{k-x}(t))$ is low, $g_k(t)$ is an informative spike, and it increases the likelihood of some $x' \neq x$. So, $P(x, t_{k+})$ should decrease. However, if $R_k(t)$ is low, $(\lambda_0 + \Delta\lambda m_{k-x}(t))$ is also low and $g_k(t)$ is not an informative spike, so $P(x, t_{k+})$ shouldn't change much.

Further, the spike from neuron k will affect our belief in the pixel value being ON at pixel location i as follows:

$$m_i(t_{k+}) = m_i(t_{k-}) + m_i(t_{k-})\frac{\Delta\lambda(1 - m_i(t_{k-}))P(x = k - i, t_{k+})}{(\lambda_0 + \Delta\lambda m_i(t_{k-}))} \tag{2.9}$$

If the displacement $x = k - i$ is likely, then its likely that neuron k fired by observing $s_i = 1$. So, $m_i(t)$ should increase. If $x = k - i$ is not likely, $P(x = k - i, t_{k+})$ is low and neuron k is unlikely to fire by observing s_i, hence $g_k(t)$ shouldn't affect $m_i(t)$. This justifies $P(x = k - i, t_{k+})$ being in the numerator in the numerator in equation 2.9. $(\lambda_0 + \Delta\Lambda m_i(t_{k-}))$ is a normalization factor.

Using these equations for every pixel i, displacement x at every timestep t, we can simultaneously infer x, b: $x(t) = \text{argmax}_x\, P(x, t)$ and $b(t) = m_i(t)$ thresholded by 0.5 at every timestep t.

2.3 Implementation and results

For our own implementation of FBD, we used the following parameters and assumptions:

1. We assume that the eye jitters in a 2-D plane, and we assume that the eye can jitter only about 20 pixels in positive or negative direction of x or y axes. At $x = (0, 0)$, the image pixels s_i would align one-to-one with the grid of ganglion cell neurons k.
2. Random walk diffusion coefficient $D = 100$ pixel$^2/s = 0.1$ pixel$^2/ms$ for simulating eye jitter trajectory. Burak et.al. state that human eye movements resemble random walks with this D.
3. To initialize the set of probabilities $P(x)\,\forall x$, we assume that the eye is stable at first and

doesn't jitter, so that $P(x = (0,0)) = 1$ and hence $P(x = (x_1, x_2)) = 0 \, \forall (x_1, x_2) \neq (0,0)$.

4. To initialize the set of probabilities $\{m_i\} \, \forall i,$, for binary images we set $m_i = 0.5 \, \forall i$ because at every pixel location, we are equally uncertain about the pixel at that location being ON or OFF.

Suppose that we are looking at a black and white image containing the letter "E" as show in Figure 2.3.

Figure 2.3: Binary image of the letter "E"

This image of the letter "E" may fall on a patch of the retina (the grid of homogeneous ganglion cells) as shown in Figure 2.4 (from [4]).

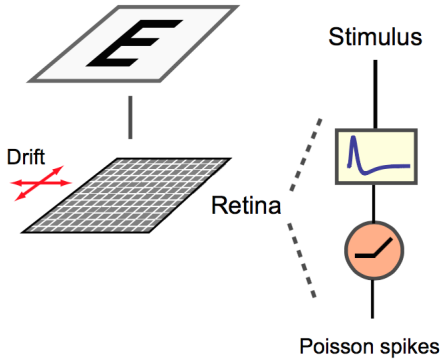


Figure 2.4: Retinal spike generation model for a binary image

The results of using FBD on spike trains generated for a duration of 300 ms simulating eye jitter with a random walk diffusion coefficient $D = 0.1$ pixel$^2/ms$ on the image in Figure 2.3 are shown in Figure 2.5 and Figure 2.6

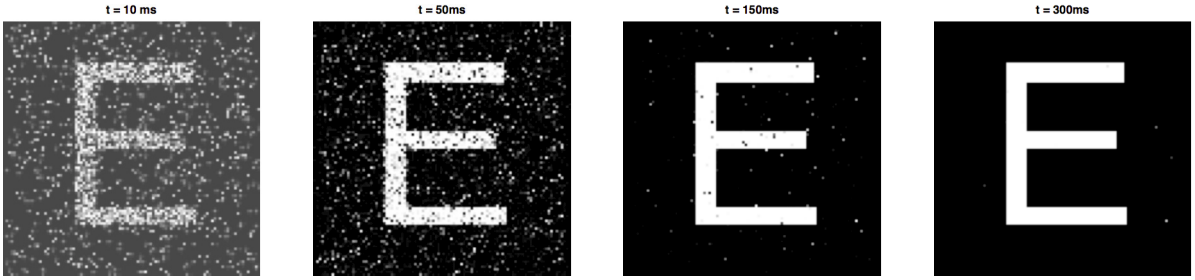


Figure 2.5: Inferred binary image over time for static image stimulus of Figure 2.3

Over time, FBD is able to correctly infer which spikes are from ON parts of the image and which spikes are from the OFF parts. Further, the trajectory predicted by FBD (in blue) is approximately equal to the trajectory of the eye jitter. In Figure 2.6, the graph of trajectory along Y-axis shows that FBD correctly infers the exact trajectory, which is why the red (original eye jitter) and blue (inferred eye jitter) overlap.

Figure 2.6: Trajectory prediction in 2-D for static image stimulus of Figure 2.3

Chapter 3

Visualizing grayscale images of static scenes

A natural question is "How does one see objects in real life?" The binary image decoder is an instructive example as it clearly demonstrates that the brain can calculate eye jitter and infer a stabilized binary image of a static object in 300 ms, the duration of one human eye fixation or the blink of an eye. But real life objects have structure and detail which may not be well captured by black or white pixels only. We now discuss how one can use FBD to reconstruct static grayscale images.

3.1 Using FBD for grayscale images

Burak et.al. presented a set of equations from which they derived a binary image decoder. These equations (stated below) can be used to make a grayscale image decoder. In image processing, we represent gray colors as values in the range $[0, 1]$. A gray image may have many values in the range $[0, 1]$. This range represents a real interval with infinite values. Let us assume that the range $[0, 1]$ is divided into l equal parts, hence in equal $(l + 1)$ levels, so that the image stimuli have gray values in one of the $(l + 1)$ levels. For example if $l = 2$, the possible gray value levels are $\{0, 0.5, 1\}$ (2 equal parts $[0, 0.5]$, $[0.5, 1]$ and 3 levels: $\{0, 0.5, 1\}$). So, we have an input image $\{s_i\} \forall$ pixels i, where pixel value at location i, $s_i \in \{v_1, v_2, \ldots, v_{l+1}\}$.

The setup for handling grayscale input is similar to that in 2.1 in the previous chapter. We

have a grid of ganglion cell neurons and each neuron fires at a firing rate as described by equation 2.1with the same constraints applied on λ_0, λ_1. Given ganglion cell spike trains $g_k(t)\,\forall t$ for all neurons k, we want to infer a stabilized grayscale image $s_i(t)\,\forall i$ and eye jitter $x(t)$ at every timestep t.

Following equation 2.2, at every pixel i, FBD should track $P_i(s_i, t)\,\forall s_i$. In grayscale images with l partitions of gray color , $s_i \in \{v_1, v_2, \ldots v_{l+1}\}$. So, while decoding a grayscale image of a natural scene with $l = 9$ gray levels for example, FBD should track $P_i(s_i = v_j, t)\,\forall j \in \{1, 2, \ldots, 10\}$

Suppose at timestep t we don't see any spikes. Then, the belief in probabilities of different displacements, $P(x, t)$, should weaken as described in equations 2.3 and 2.4.

Further, the probabilities of every pixel being of any specific color should also diffuse depending on what is the expected color at that pixel:

$$\frac{dP_i(s_i = v_j, t)}{dt} = (\rho_i(t) - \lambda(v_j))P_i(s_i = v_j, t) \tag{3.1}$$

where v_j is a grayscale value, s_i is the pixel observed at location i and where $\rho_i(t)$ is the expected firing rate at pixel location i at timestep t,

$$\rho_i(t) = \sum_{v_j \in \{v_1, v_2, \ldots v_{l+1}\}} \lambda(v_j)P_i(s_i = v_j, t) \tag{3.2}$$

So the updated value of $P_i(s_i = v_j, t)$ is:

$$P_i(s_i = v_j, t_+) = P_i(s_i = v_j, t_-) + (\rho_i(t) - \lambda(v_j))P_i(s_i = v_j, t_-) \tag{3.3}$$

$\rho_i(t)$ is value which denotes expected color at pixel location i because by definition, the value of $\rho_i(t)$ will be affected most by the most probable grayscale color value. Intuitively, the update works as follows:

1. If $\rho_i(t) > \lambda(v_j)$, then $\frac{dP_i(s_i = v_j, t)}{dt}$ will be positive, and $P_i(s_i = v_j, t)$ will increase. This implies that we expected a high firing rate but the rate $\lambda(v_j)$ is low, and in the absence of

spikes we increase our belief in expecting a lower firing rate (and hence darker gray value) at pixel location i.

2. If $\rho_i(t) < \lambda(v_j)$, then $\frac{dP_i(s_i=v_j,t)}{dt}$ will be negative, and $P_i(s_i = v_j, t)$ will decrease. This implies that we expected a lower firing rate at pixel location i, and in the absence of spikes our belief in expecting a high firing rate (and hence lighter gray value) at pixel location i will decrease (Recall that absence of spike at timestep t denotes lack of observation of a bright color at t).

3. If $\rho_i(t) = \lambda(v_j)$, then $P_i(s_i = v_j, t)$ won't change at all, as our confidence in the hypothesis, expected firing rate at pixel i shouldn't change in this case.

Suppose that at timestep t, we observe a spike from ganglion cell neuron k ($g_k(t) = 1$). We can use this spike at k to update $P(x, t)$ for all x as follows:

$$P(x, t_{k+}) = P(x, t_{k-}) \frac{\rho_{k-x}(t_{k-})}{R_k(t_{k-})} \tag{3.4}$$

where

$$R_k(t) = \sum_x \rho_{k-x}(t) P(x, t) \tag{3.5}$$

where t_{k-} indicates the time before updating a quantity due to a spike from neuron k, t_{k+} indicates the time after updating a quantity due to a spike from neuron k.

The only difference in equations 3.4, 3.5, from 2.7, 2.8 respectively is the use of expected firing rate $\rho_i(t)$. In fact, for the case of binary images, one can derive 2.7, 2.8 using 3.4, 3.5. Hence, the intuition behind the updates described here is similar to that described earlier in equations 2.7, 2.8.

Also, our belief in probabilities of different grayscale colors at pixel location i can be calculated as:

$$P_i(s_i = v_j, t_{k+}) = P_i(s_i = v_j, t_{k-}) + P_i(s_i = v_j, t_{k-}) \frac{(\lambda(v_j) - \rho_i(t_{k-}))P(x = k - i, t_{k+})}{\rho_i(t_{k-})} \tag{3.6}$$

Again, this equation differs from equation 2.9 by using $\rho_i(t)$. Note that in 3.6, we use $(\lambda(v_j) - \rho_i(t_{k-}))$, the sign of this quantity is reversed in 3.3. This sign reversal is necessary as in equation 3.6, we update the confidence in our hypothesis about different grayscale color values of the image stimulus given the spike from a neuron k. So the update will work as follows:

1. If $\rho_i(t_{k-}) > \lambda(v_j)$, then $(\lambda(v_j) - \rho_i(t_{k-}))$ will be negative, and $P_i(s_i = v_j, t)$ will decrease. This implies that we expected a high firing rate but the rate $\lambda(v_j)$ is low, and in the presence of spikes we decrease our belief in expecting a lower firing rate (and hence darker gray value) at pixel location i.

2. If $\rho_i(t_{k-}) < \lambda(v_j)$, then $(\lambda(v_j) - \rho_i(t_{k-}))$ will be positive, and $P_i(s_i = v_j, t)$ will increase. This implies that we expected a lower firing rate at pixel location i, and in the presence of spikes our belief in expecting a high firing rate (and hence lighter gray value) at pixel location i will increase (Recall that a spike at timestep t denotes the possibility of observation of a bright color at t).

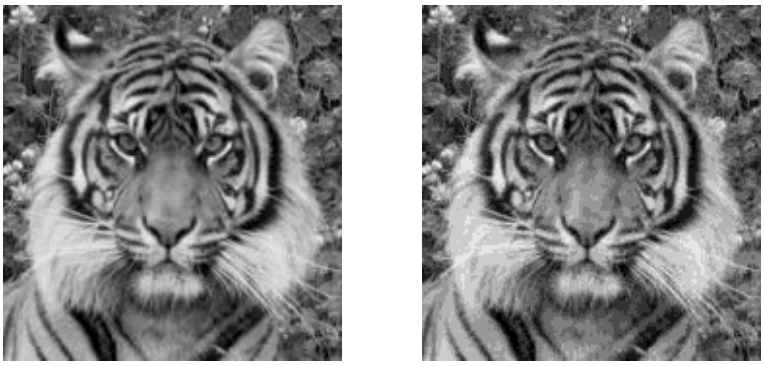
3. If $\rho_i(t_{k-}) = \lambda(v_j)$, then $P_i(s_i = v_j, t)$ won't change at all, as our confidence in the hypothesis, expected firing rate at pixel i shouldn't change in this case.

Using this set of equations, we can use FBD to infer eye jitter $x(t)$ and stabilized image $\{r_i\} \forall i$ of a static grayscale scene simultaneously: $x(t) = \text{argmax}_x P(x, t)$ and

$r(t) = \sum_{v_j \in \{v_1, \ldots v_{l+1}\}} v_j P(s_i = v_j, t)$ at every timestep t.

3.2 Implementation and results

Consider a grayscale image of a tiger from [1]. Figure 3.1 shows the original input image, and a version of the image with only 10 grayscale colors. The figure also shows how FBD is able to estimate the tiger's image over time. The inferred trajectory over time is presented in Figure 3.2.



Original Tiger Grayscale Image Discretized to 10 Gray colors

Inferred tiger image using grayscale FBD over time

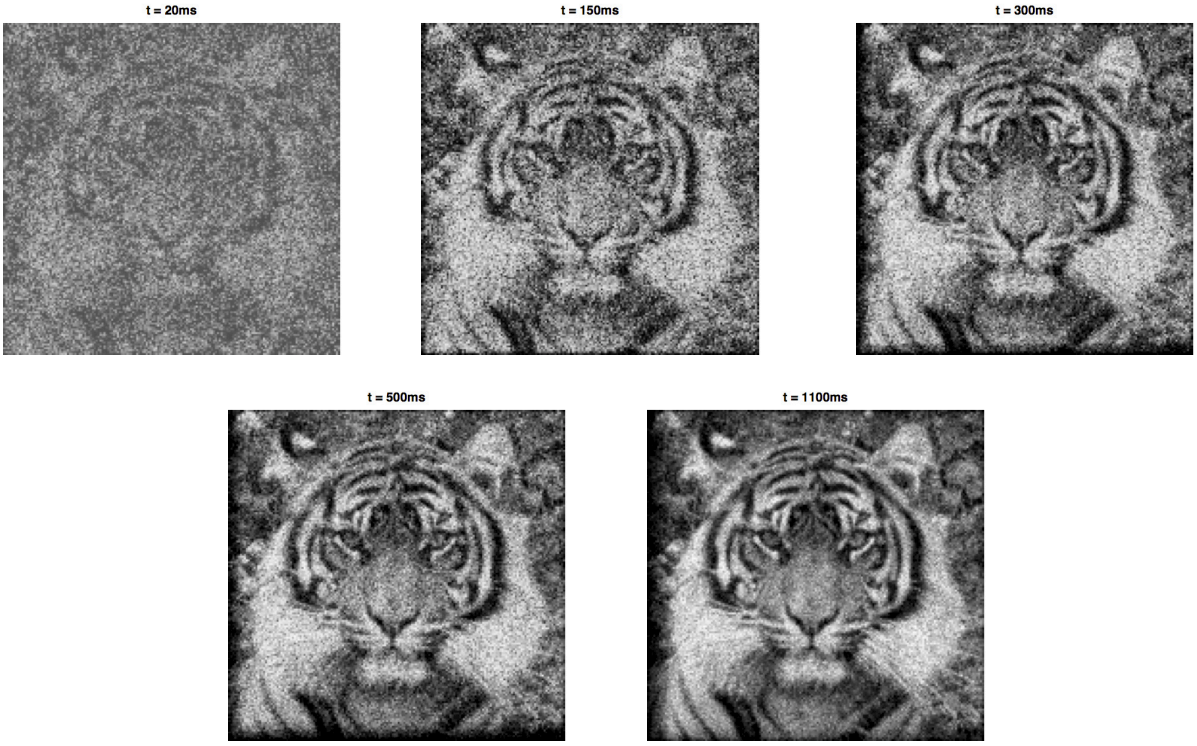


Figure 3.1: Tiger input image, and inferred image via FBD

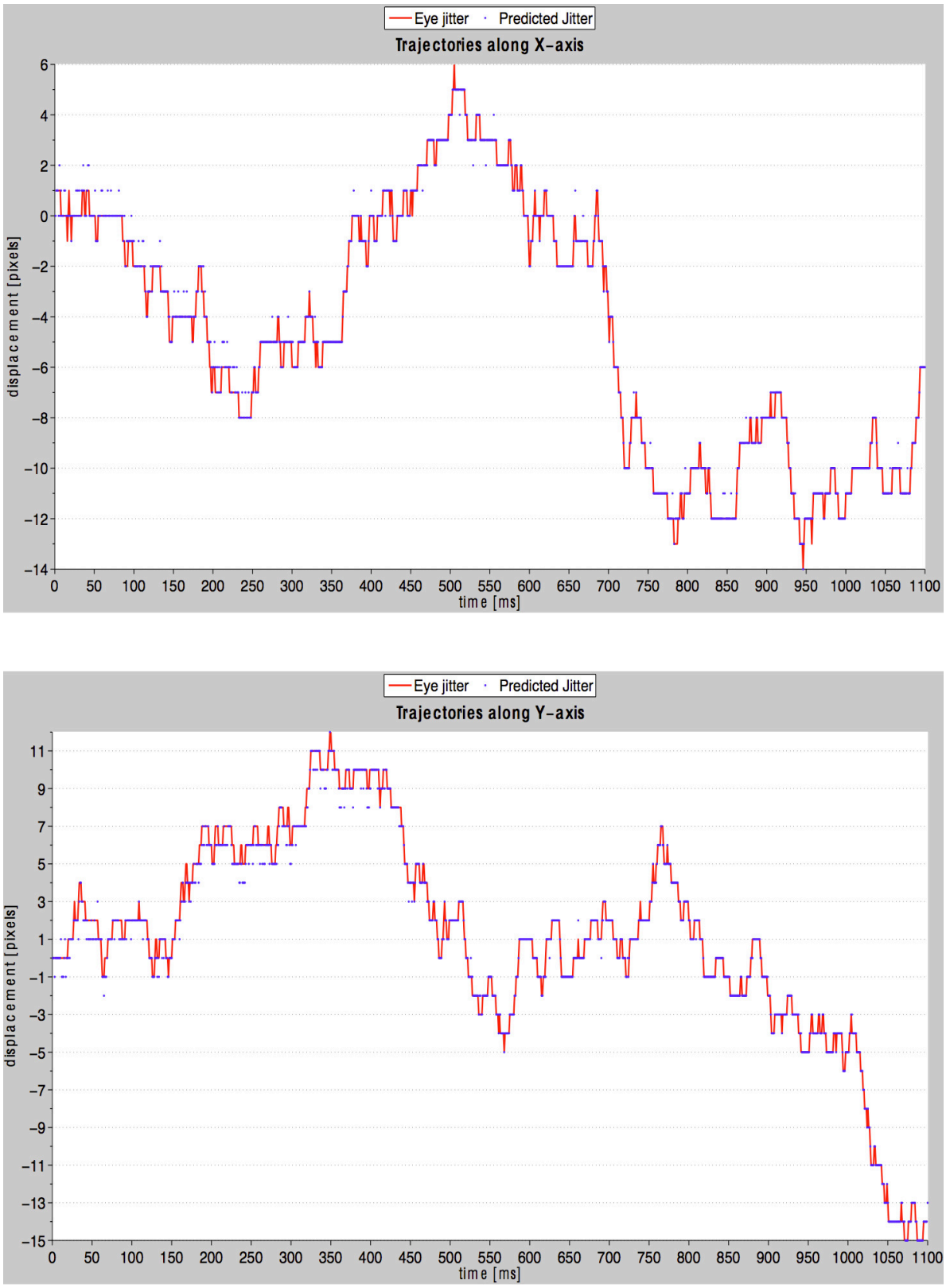


Figure 3.2: Trajectory prediction in 2-D for static image stimulus of Figure 3.1

Chapter 4

Object reconstruction using FBD

FBD can be used to explain how the cortex can infer a static scene using retinal spike trains. However, we live in a dynamic world where objects move a lot, sometimes as clearly as an aeroplane flying across the sky, or with motions subtle enough to deceive human eyes, such as that of a plucked guitar string. Further, there are many different kinds of motion possible in 2-D space. An object as simple as a ball may fall in the air in a linear fashion or in a trajectory shaped like an arc, such as when a cannonball is fired at an angle. There are objects that move partly so that they are fixed w.r.t. a pivot point, such as a pendulum in a grandfather clock or a spinning wheel in a gym's stationary bicycle. And objects can consist of many parts themselves that move in a complicated fashion, such as a bird flying by flapping its wings or a pedestrian walking on a street using leg movements. Further, there can be multiple objects moving in a scene, such as many people crossing a street, or horses racing in a field.

With so many possible object motions and motion details that change rapidly, its remarkable how the few spikes generated by the retina on viewing these objects momentarily in different positions can help us to see what the object is and identify it, along with its trajectory of motion. Naturally, the amount of light that falls on the moving object and the brightness of its background affect its visibility. So, for building a system that can infer and reconstruct moving objects, we first examine how we can work with binary scenes with white objects (static or dynamic) and black background, an easier case than a grayscale natural scene. Also, we consider scenes with

one object moving in a linear fashion in 2-D space.

4.1 Stimulus with black background

Consider that we have a video of a check patterned black and white ball moving across a blank frame linearly. Given this video sequence $\{s_i(t)\}\,\forall i$ for every timestep t for a duration of T ms, we first generate spike trains for every neuron k, $g_k(t)\,\forall t$ at every timestep t. Suppose while generating these spike trains, we don't simulate eye jitter, so that the spikes are only generated from the video stimulus. Our goal is to infer the trajectory of the moving object, $x(t)$ for every timestep t, and reconstruct the object in one location using spikes from the object collected over time.

An object moving in a linear fashion will appear to us as simply moving continuously from one pixel location to another. In order to reconstruct the object, we need to know at every timestep t, which spikes $g_k(t)$ are from neurons observing the object. Since the object is moving across the background, we may view this motion as global motion of the first frame $f(t) = \{s_i(t)\}\,\forall i$ containing the object. Therefore, for such a stimulus, the trajectory of the object can be inferred as a set of displacements of the frame $f(t)$ over time. This is similar to inferring eye jitter trajectory. And inferring a stabilized binary image is the job of FBD.

The stimulus of a ball moving towards right is shown in Figure 4.1, and how the retina may infer the object of the ball is shown in Figure 4.2. In Figure 4.2, the retina is shown as a red grid patch of neurons that seem to jitter towards the right, causing one to view the ball moving towards the right.

In this case, since the object motion trajectory is inferred as global frame jitter, the spikes from white areas of the check patterned ball will align and combine as described in Chapter 2, to give us a coherent reconstructed check patterned black and white ball as presented in the stimulus.Therefore, when we simulate eye jitter while generating the spike trains, the displacements inferred by FBD will be combined displacements of the frame and of the trajectory of the ball.

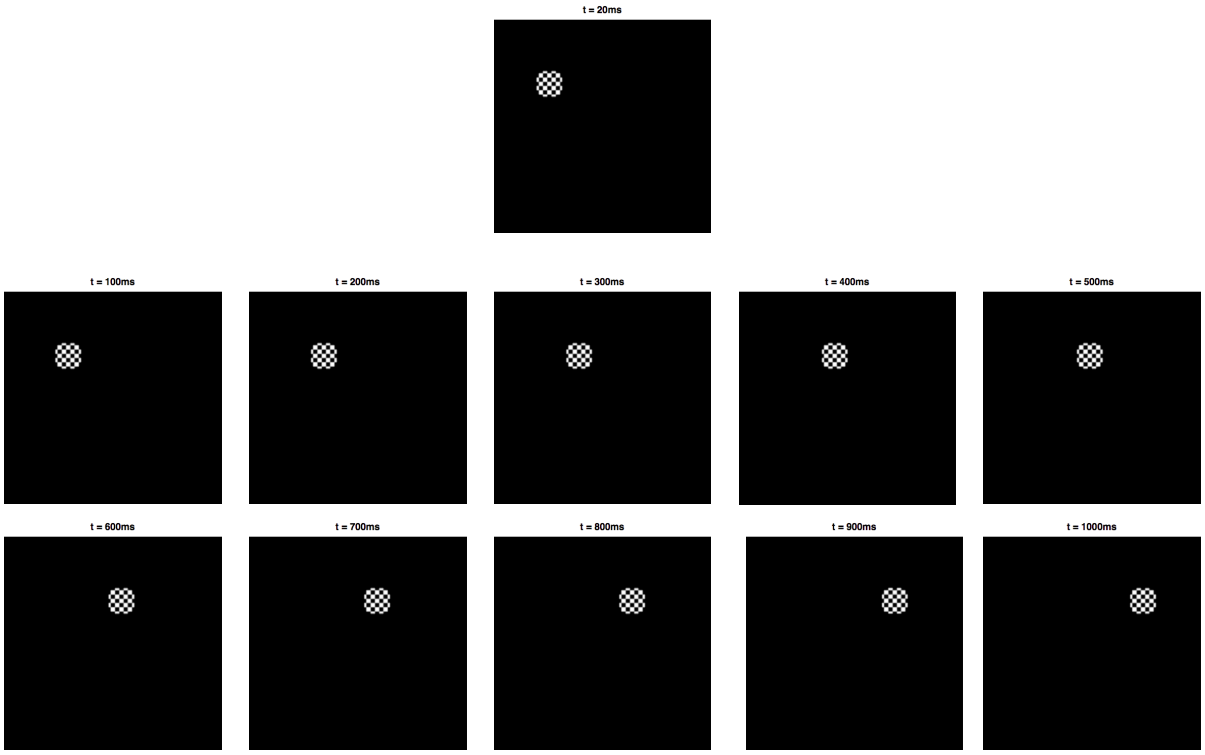


Figure 4.1: Input stimulus with a check patterned ball against a black background

However, one must note that the displacements possible, that is, the set of all x, is even longer and bigger because these displacements account for object motion across the frame. For an $m \times n$ ganglion cell grid, displacements in the x-axis can be in range $[-n, n]$ and displacements in the y-axis can be in range $[-m, m]$, as the object may move across the frame entirely. So the number of possible 2-D displacements is now $2m \times 2n = 4mn$, which increases as the size of the patch of retina or size of the visual stimulus under consideration increases.

The inferred trajectory for the stimulus shown in Figure 4.1 along with simulated eye jitter is shown in Figure 4.3. Note that in the Y direction, the ball doesn't move at all so the motion predicted by FBD is that of the eye only. In the X direction, the green line shows the motion of the ball, the red line shows the motion of the eye only, and the magenta line shows the combined displacement of the ball and eye. FBD correctly infers the combined trajectory, shown in blue.

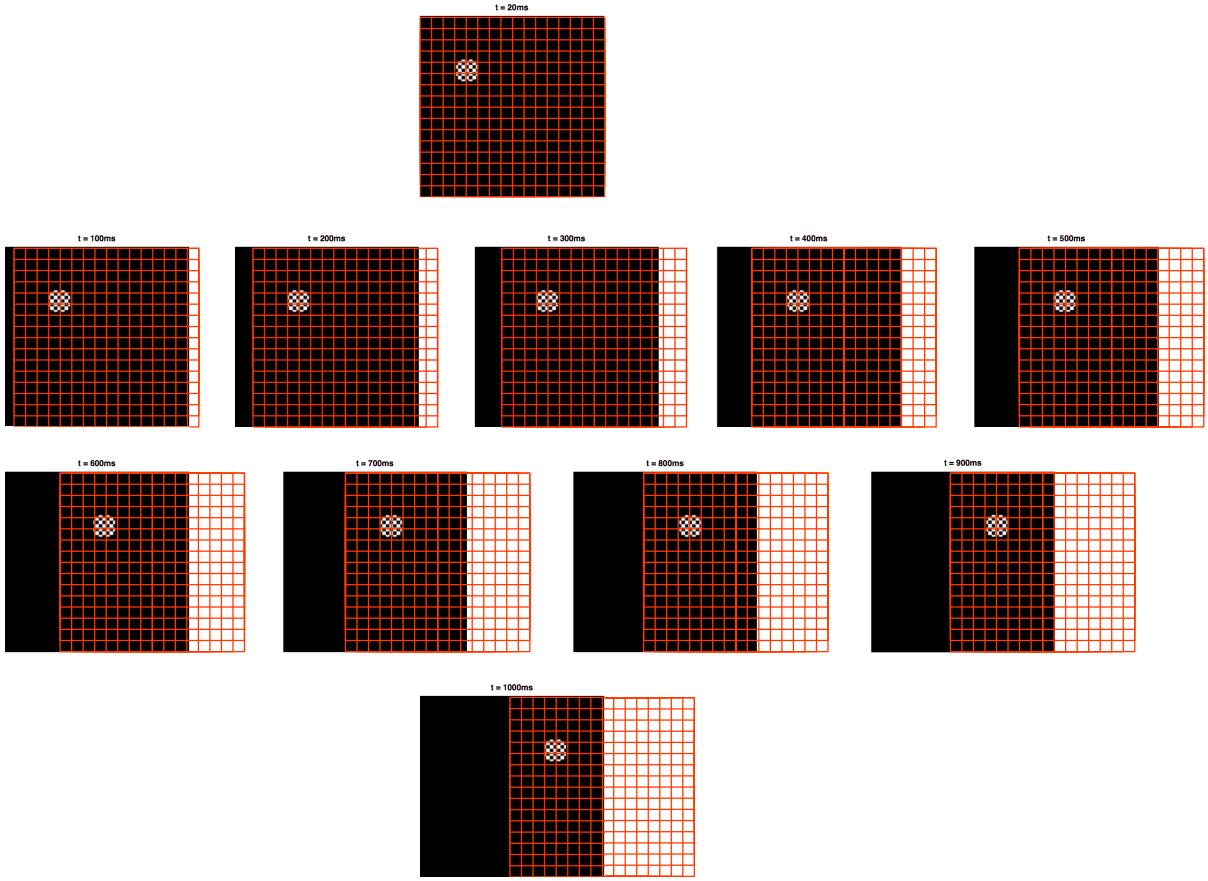


Figure 4.2: Modeling of ball motion in Figure 4.1 as a global image jitter

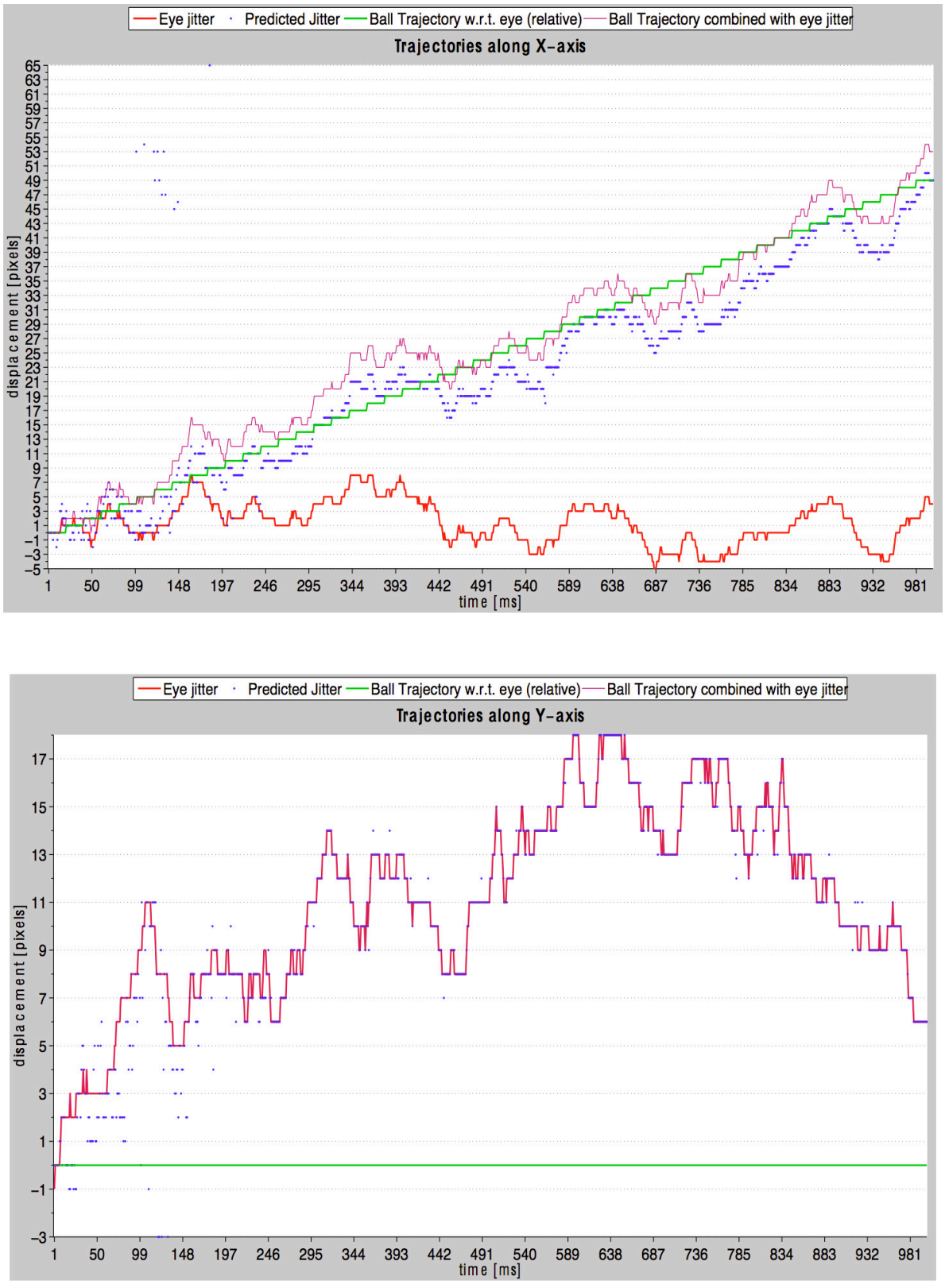


Figure 4.3: Inferred trajectory of jittered stimulus of Figure 4.1

4.2 Stimulus with static objects in background

Consider that we have a video of a check patterned black and white ball moving linearly across a frame containing a stationary letter "A" in white pixels against a black background. Given this video sequence $\{s_i(t)\}\,\forall i$ for every timestep t for a duration of T ms, we first generate spike trains for every neuron k, $g_k(t)\,\forall t$ at every timestep t. Suppose while generating these spike trains, we don't simulate eye jitter, so that the spikes are only generated from the video stimulus. Our goal is to infer the trajectory of the moving object, $x(t)$ for every timestep t, infer the objects visible in the background and reconstruct the object in one location using spikes from the object collected over time. The stimulus is shown in Figure 4.4.

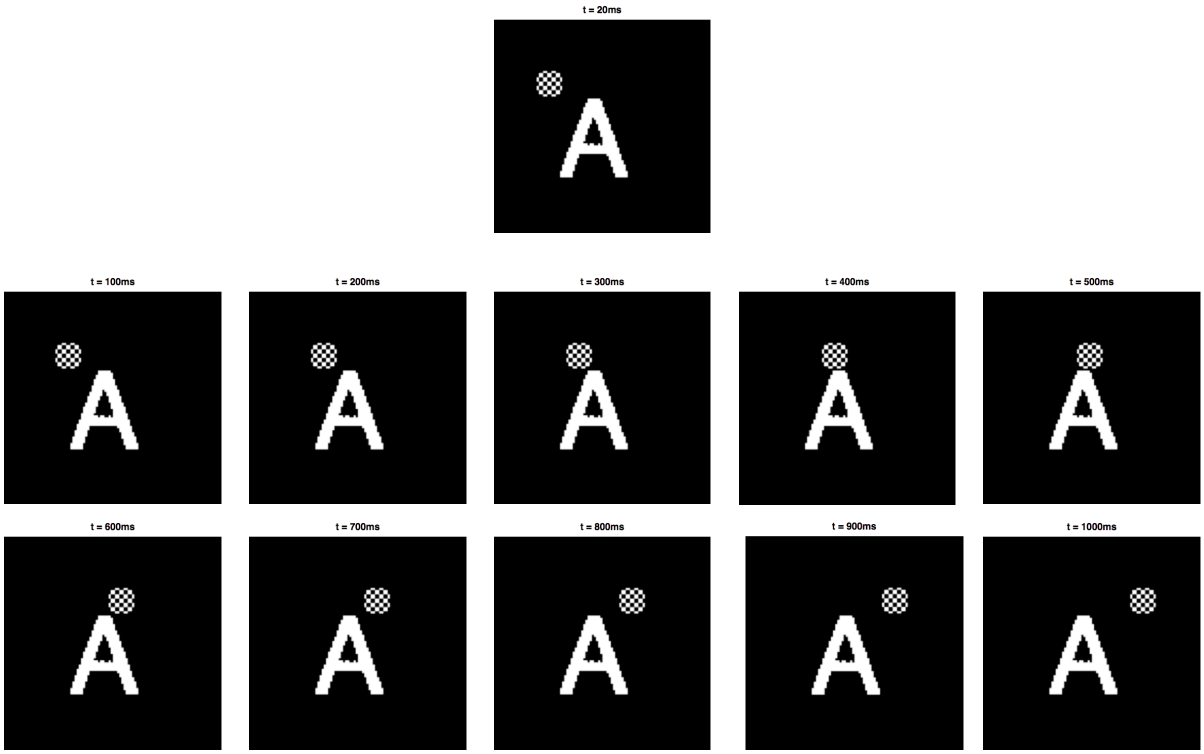


Figure 4.4: Input stimulus of check patterned ball moving against black background, containing a white "A" letter object

In this stimulus, the situation seems complicated by the presence of a static object in the background, the letter "A". The letter is represented with white pixels, which means that over a

duration of T ms, neurons observing pixels from "A" will fire a lot of spikes. Suppose that we run FBD on the spike trains $g_k(t)\,\forall k, t$ generated from this stimulus. One may expect that the spikes from the ball will confuse the decoder and cause it to infer the stimulus incorrectly. But, lets examine what will happen.

As the ball moves across the frame, neurons observing white segments of the ball will fire spikes. However, a particular set of neurons which fire spikes for the ball will only do so for the short duration the ball is visible to those neurons. As the ball moves away, the neurons will observe a black background and will fire fewer spikes over time for the remaining duration of the stimulus. For initial inference, we may expect the spikes from the white regions of the ball to confuse FBD as the object is static for a while and even when it moves (depending on its speed of movement), the parts of the object may overlap with the previous location of the ball. However, as time progresses and the ball moves away, neurons in those regions will stop emitting spikes. Two things happen:

1. As stated in equation 2.6, our belief in observing a white pixel diffuses over time due to absence of spikes, in regions where the ball was not present initially. By the time the ball reaches a location farther away than its starting position, FBD may already have strong belief that neurons in those regions are observing black pixels (due to 2.6)and hence the few spikes emitted by viewing the ball in that region will be disregarded.
2. Initially where the ball was located, the belief that there are neurons in that region observing white pixels will weaken over time as the ball moves away, because of diffusion in the probability due to equation 2.6.

This means that at the end of the computation, FBD will return a stabilized image of the background, that is, objects static in the scene and will ignore any spikes from the object. The results of the first run of FBD are shown in Figures 4.5 and 4.6.

One may question, why doesn't the ball's motion trajectory affect the interpretation of a stabilized background? Since the spikes from the ball are effectively ignored as time progresses

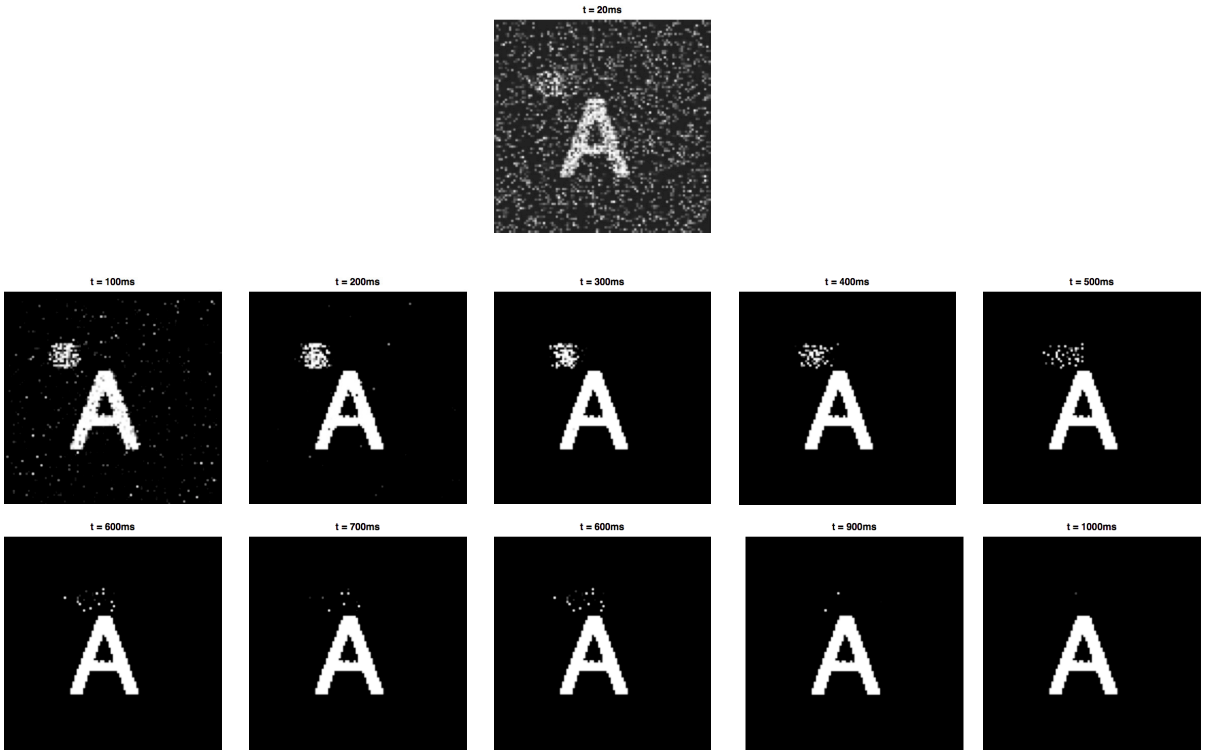


Figure 4.5: Inferred background image after first run of FBD on stimulus described in Figure 4.4

any belief in displacements x that track the ball trajectory will weaken as well, due to equation 2.4.

Given a stable background image $\{sb_i\}\,\forall i$, we need to pave a way to identify and collect spikes from neurons observing the object. At every timestep t, for every neuron k, we observe a spike if $g_k(t) = 1$. The spike is from the object only if its not from the background. One way to solve this problem is that at every timestep t, we use the inferred global jitter $x(t)$, and match $g_k(t)$ with $sb_{i=k-x}$. If $sb_{i=k-x}$ is ON, pixel i belongs to a white background object and if $g_k(t) = 1$, the spike is from a background object and hence its not a part of the moving object. But if $g_k(t) = 1$ and $sb_{i=k-x}$ is OFF, pixel i probably belongs to the object. This is probable only because sometimes neurons emit spikes by observing black pixels too. In this way, at every timestep t and for every neuron k, we can identify spikes that are probably part of the foreground moving object (in our case the check patterned ball). Spikes from the black background are few

in number and over time are effectively ignored by FBD.

Once we have a set of spike trains that are from the moving object, the situation is exactly the same as described in section 4.1. We reuse FBD, with a larger set of possible displacements and infer the ball's motion trajectory and a stabilized image of the ball.

Again, when we simulate the eye jitter while generating spike trains, the eye jitter will be interpreted correctly in the first step, when FBD infers a stabilized background image, because it ignores spikes and hence displacements of the moving object. The object's motion trajectory will correctly be inferred in the second step, after filtering object spikes, as described in section 4.1.

Results after running FBD the second time, using a stable background image from the first run containing the white letter "A" object to remove spikes that are from the background are shown in Figures 4.7 and 4.8.

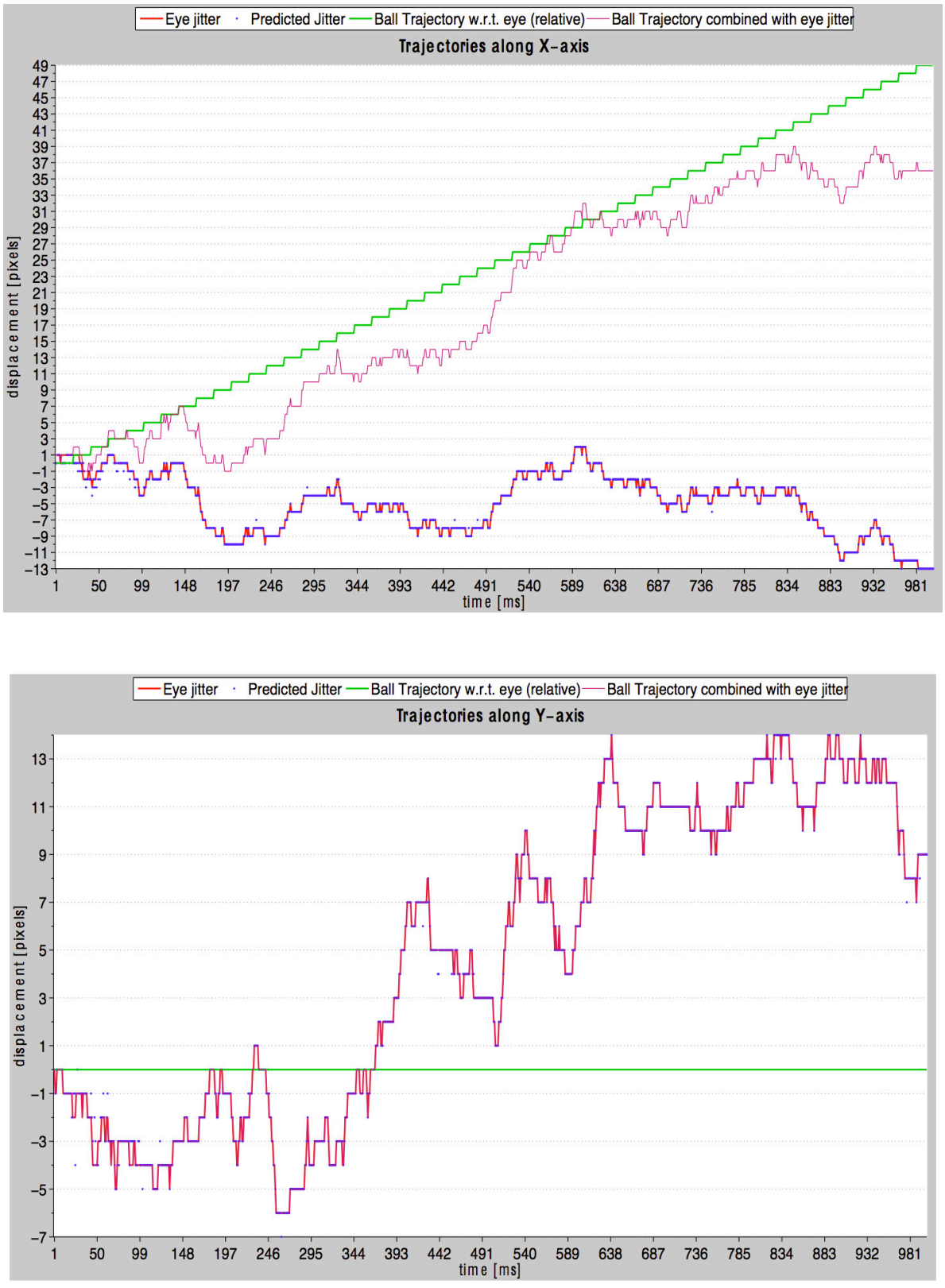


Figure 4.6: Inferred eye jitter trajectory after first run of FBD on stimulus described in Figure 4.4

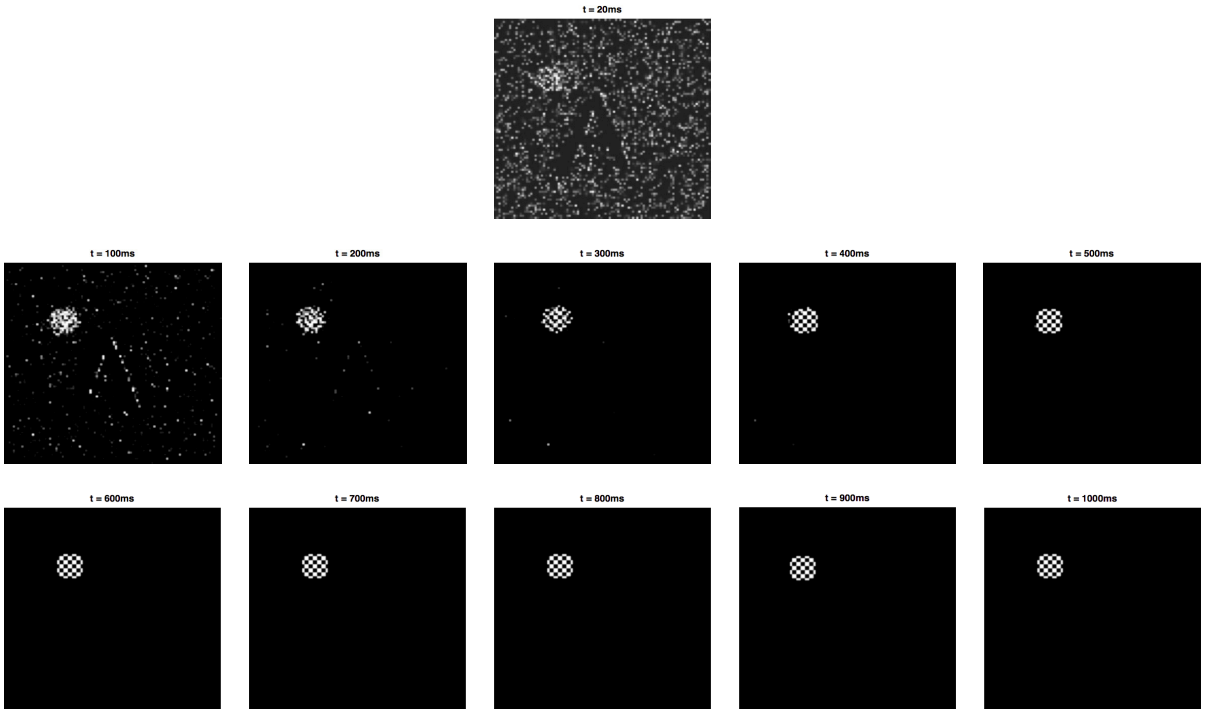


Figure 4.7: Inferred object image after second run of FBD on stimulus described in Figure 4.4

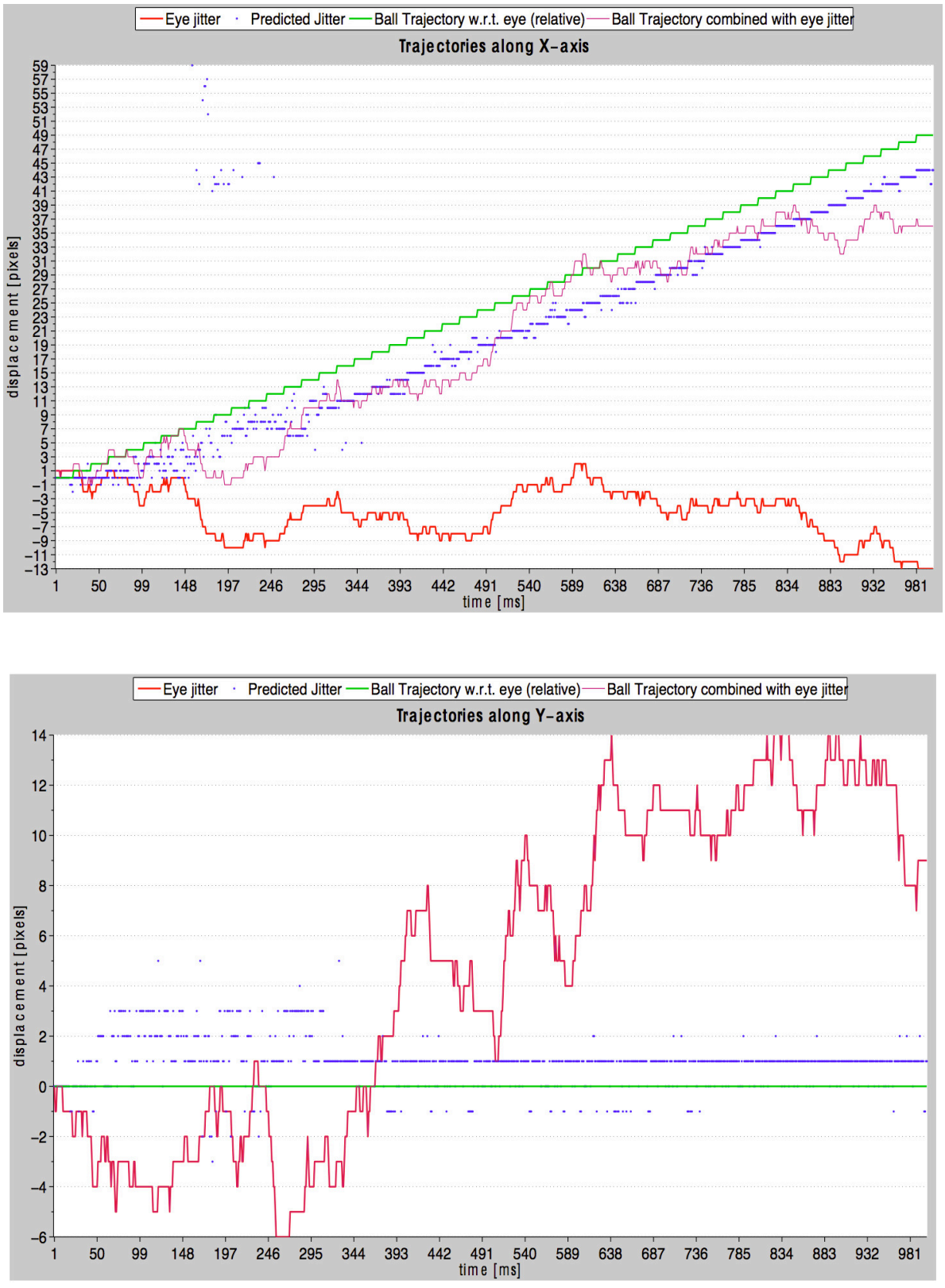


Figure 4.8: Inferred eye jitter trajectory after second run of FBD on stimulus described in Figure 4.4

4.3 Moving object reconstruction system

So, the steps to infer and reconstruct an object moving linearly in 2-D space are:

1. Infer the eye jitter $x(t)$ and stabilized background image $\{sb_i\}\,\forall i$ using FBD.
2. Use $\{sb_i\}\,\forall i$ to filter out spikes that are produced by neurons observing the object moving.
3. Using the set of filtered object spikes, use FBD to collect object spikes and reconstruct the object in one place.

Chapter 5

Visualizing moving objects in natural scenes

To use the system proposed for object reconstruction of moving objects in grayscale images of dynamic natural scenes, we can use a grayscale version of the FBD described in detail in Chapter 3.

5.1 Artificial Stimulus in Grayscale

Before dealing with complexities in natural scenes, we first convert the stimulus of a check patterned ball moving linearly across a background containing a static "A" letter to grayscale so that "A" is white, the background is gray and the ball is black and white check pattern.

According to equation 2.1, we construct a stimulus using firing rates $\lambda_0 = 10$ Hz, $\lambda_{0.5} = 55$ Hz, $\lambda_1 = 100$ Hz for gray colors $0, 0.5$ and 1 respectively. Since we track the probability of each gray color when we use spikes to update $P_i(s_i = v_j, t) \, \forall v_j \in \{0, 0.5, 1\}$, one may assume that using FBD as described in the last chapter, one can reconstruct a stable gray background with a white "A" on it, and then reconstruct the check patterned ball. However, one important point to consider is that the background on which the ball moves is no longer black but gray, and hence will fire more spikes (about 5.5% of the duration of the stimulus) than the black background. Since the object moves across the gray background, the few spikes from the object in different positions will be ignored. The entire frame region where the object moves will be estimated

as gray in color. So, we may correctly recover a stabilized image of the background. However, when we try to identify spikes from the moving object, the stabilized background will indicate all spikes are from the background because the spikes from the object will be ignored in comparison to the spikes from the gray background. Therefore we need to have a more reliable method to use a stabilized background image of any grayscale color value(s) to identify spikes from the moving object.

We know that to generate a spike train of a firing rate $\lambda(v)$ specific to a color v, for a stimulus of duration T ms, we sample spike times from a Poisson distribution of mean $\lambda = \lambda(v) * T$. For figuring out which spikes belong to the object, we can use the following approach: When we compute the stabilized background $\{bg_i\}\,\forall i$, we also recover the jitter trajectory $\{x(t)\}\,\forall t$ of the eye. We can shift each frame of the stimulus containing spikes $\{g_k(t)\}\,\forall k$ by $\{-x(t)\}$ at every timestep t to get a shifted stimulus of spike trains, $\{g_{i'}(t) = g_{k-x(t)}(t)\}\,\forall k, t$. Now each spike at $i' = k - x(t)$ aligns with pixel location i in the stabilized background image $\{bg_i\}\,\forall i$, so that $i' = i$.

For every Δt frames in the shifted stimulus, i.e. for all $\{gs_i(t), gs_i(t+1), \ldots gs_i(t+\Delta t - 1)\}\,\forall i$:

1. Compute the total spike count c at every location i over the time period t to $t + \Delta t - 1$, i.e., $\{c_i\}\,\forall i$.
2. For every $t' \in \{t, \ldots, t + \Delta t - 1\}$:
   (a) Compute a background probability map pB such that $pB_i(t') = P(c_i, \lambda(bg_i) * \Delta t)$ where $P(X, \lambda)$ represents the Poisson probability distribution function with mean parameter λ for a given value X.
   (b) Compute a foreground mask F using $B = pB > bthresh\,\forall i$, and then $F = 1 - B$, where $bthresh$ is a threshold value one can infer empirically.
   (c) Modify spike input frame $gs_i(t')$ to get a new spike input frame $gf_i(t') = gs_i(t') * F$ where $gf_i(t')$ contains the spikes which are from the moving object.

Δt, the number of frames to evaluate in every iteration , is also a value that can be inferred empirically. However, we know that a lot of movies are projected at frame rates between of 24 Hz to 48 Hz. This implies that the eyes can see objects clearly when they move at 1 pixel$/40$ ms to 1 pixel$/20$ ms respectively. When we present stimuli with objects moving faster than 1 pixel$/\Delta t$ ms, spikes from moving objects may not align in the same pixel locations when we count the spikes over time period of Δt as we won't sample spike counts fast enough. So for our experiments we use a value of $\Delta t = 20$. At $\Delta t = 20$ ms, $\lambda(0) = 0.01/$ms, we expect only an average of 2 spikes from a black pixel. By decreasing Δt, we will lose the ability of using a Poisson probability function for a valid number of spikes. That is, if we try to sample faster, our scheme won't work well for darker gray values.

We now have a scheme to deal with any background color when we identify which spikes are from the moving object.

There is another potential problem: When we track pixel value probabilities, $P_i(s_i = v_j, t)\,\forall i, t$, colors that look similar can confuse the decoder, especially when we rely on spike count because similar color values have similar firing rates. One way to deal with this is to extend FBD to use two kinds of neurons, ON and OFF neurons what have firing rates defined as:

$$\lambda_{on}(v) = \begin{cases} \lambda_S, \text{if } v < 0.5 \\ 2(\lambda_H - \lambda_L)x + (2\lambda_L - \lambda_H), \text{if } v \geq 0.5 \end{cases} \tag{5.1}$$

for an ON neuron, and

$$\lambda_{off}(v) = \begin{cases} \lambda_S, \text{if } v > 0.5 \\ 2(\lambda_L - \lambda_H)x + \lambda_H, \text{if } v \leq 0.5 \end{cases} \tag{5.2}$$

for an OFF neuron. Here, λ_S is a very low value to indicate the fewest non-zero number of spikes, λ_L and λ_H are similar to λ_0, λ_1, so that the neurons fire at firing rates linearly proportional to the grayscale color value. We use $\lambda_S = 0.001$/ms, $\lambda_L = 0.01$/ms, and $\lambda_H = 0.1$/ms respectively in our experiments. A plot of the firing rates for ON and OFF neurons is shown in Figure 5.1.

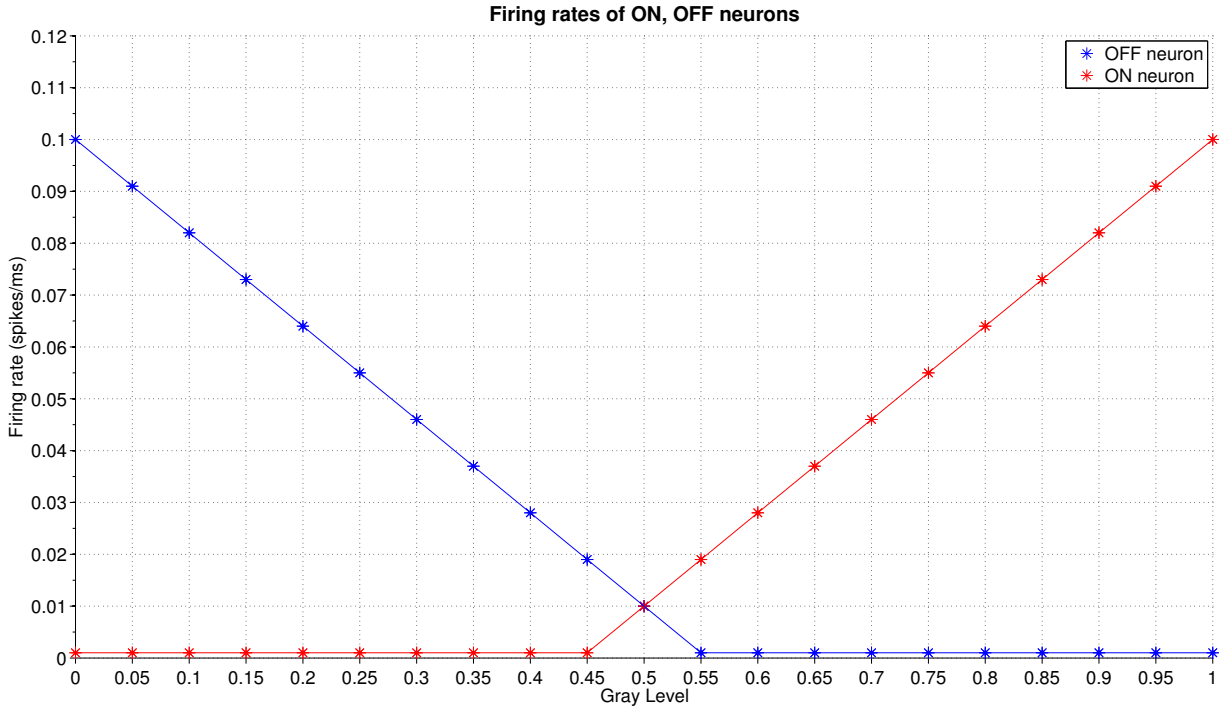


Figure 5.1: Neuron Firing Rates

So we can generate spike trains $\{g\_on_k(t)\}\ \forall k, t$ and $\{g\_off_k(t)\}\ \forall k, t$ with eye jitter $\{x(t)\}\ \forall t$. For each stimulus, we can run FBD individually to recover background and foreground sequentially. Our final procedure is as follows:

1. Run FBD for $\{g\_on_k(t)\}\ \forall k, t$ to recover jitter trajectory $x(t)$ and stabilized background $\{bg\_on_i\}\ \forall i$. Note that here $bg\_on_i = \sum_{v_j} P_i(s_i = v_j, t)v_j$. Similarly, $\{bg\_off_i\}\ \forall i$, where $bg\_off_i = \sum_{v_j} P_i(s_i = v_j, t)v_j$ as well.
2. Run FBD on shifted stimuli $\{gs\_on_i(t)\}\ \forall t$ and $\{gs\_off_i(t)\}\ \forall i, t$ to reconstruct the object using ON and OFF stimuli respectively.

One important point here is that we will no longer necessarily recover the background in the first run of FBD. In the case when the background has colors ≥ 0.5, ON cell spike trains will help recover a stable background, but the OFF cell stimuli when run through FBD will recover things in the scene that have color values ≤ 0.5, such as the black parts of the check patterned ball. In the ON Cell case, we can use the stabilized background to detect and recover the object.

In the OFF cell case, a second run of FBD may not be fruitful.

Results of using ON, OFF neurons for FBD in the object reconstruction procedure for the artificial grayscale stimulus (with jitter) (stimulus shown in Figure 5.2), are shown in Figures 5.3, 5.5, 5.7, 5.9.

Figure 5.2: Artificial stimulus input frames at different timesteps

As shown in Figure 5.3, in the first run of FBD we can correctly infer the stabilized background image at approximately 700 ms. The blue trajectories inferred by the FBD in Figure 5.4 match correctly with the simulated eye jitter trajectory, as FBD ignores the moving ball entirely. For the first 100 ms, FBD correctly infers the image of the ball using OFF neurons as shown in figure 5.5, but after that the image of the ball is less coherent. Since the ball moves towards right, FBD correctly infers that the entire global frame is moving towards left, as it ignores the spikes coming from the white A background object. Also, the right half of the inferred image is white because for OFF neurons fewer spikes indicate a high color value, and since FBD believes that the frame is moving leftwards, there is no input of spikes on the right side to decrease the probability of seeing white on the right. However, the inferred horizontal trajectory suddenly increases after 100 ms from approx. 3 pixels to 71 pixels. This jump is because the ball moved

towards the right, and it is sudden probably because after 100 ms, there aren't many spikes from the background to indicate the presence of the "A". When we lose spike information about the "A", we lose any fixed background point to measure the displacement of the ball, which probably confuses FBD.

The second run of FBD is also successful in recovering the ball and ball trajectory as from the ON neuron stimulus as shown in Figures 5.7 and 5.8 respectively. But the second run of FBD is not very informative for inferring the ball's structure or the trajectory of the image.

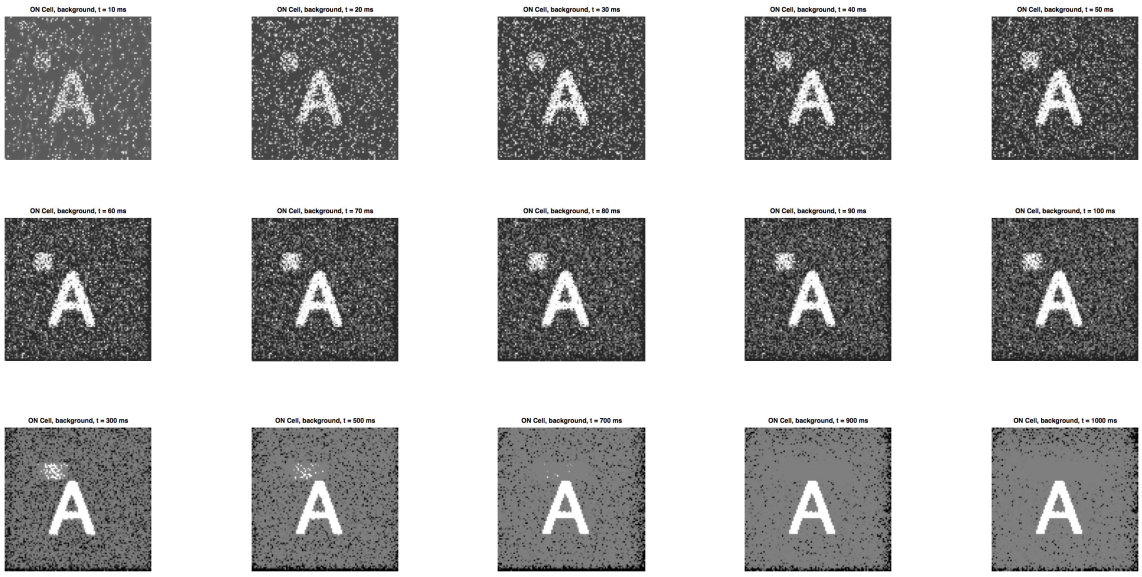


Figure 5.3: Inferred background grayscale image using ON cell firing rates, after first run of FBD on artificial stimulus

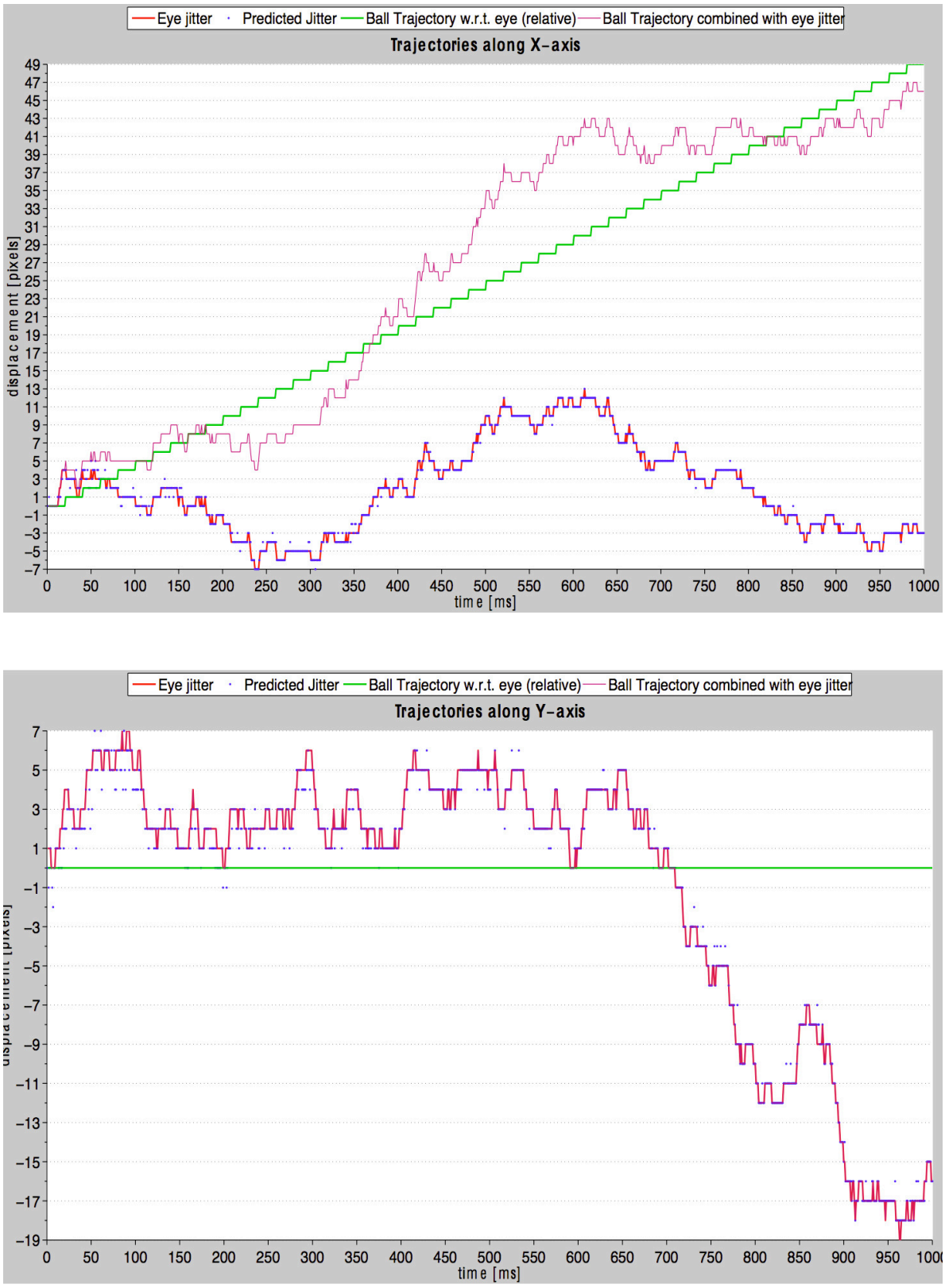


Figure 5.4: Inferred trajectories using ON cell firing rates, after first run of FBD on artificial stimulus

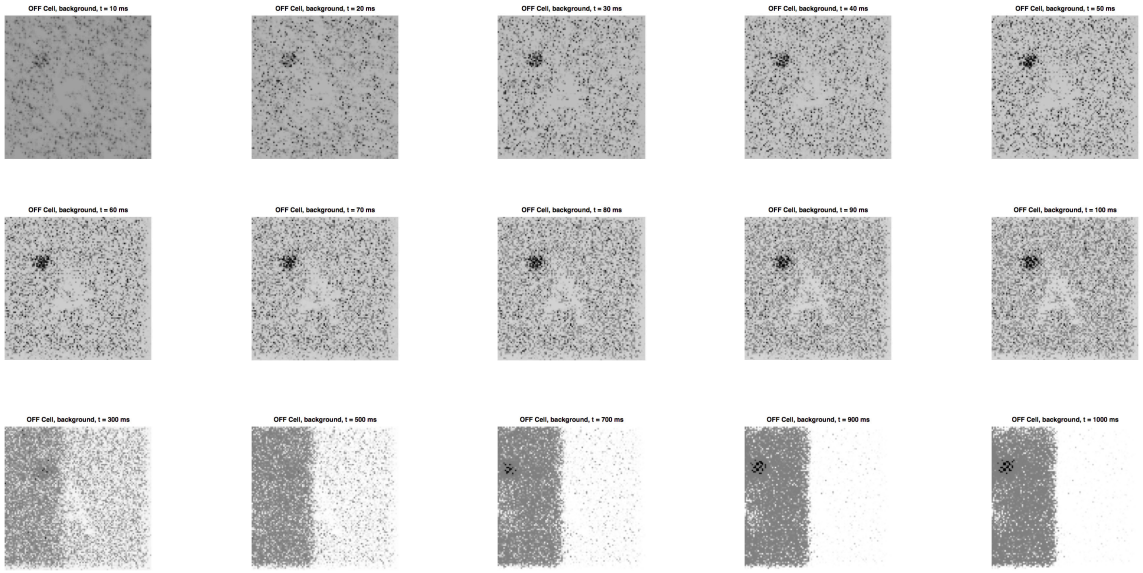


Figure 5.5: Inferred background grayscale image using OFF cell firing rates, after first run of FBD on artificial stimulus

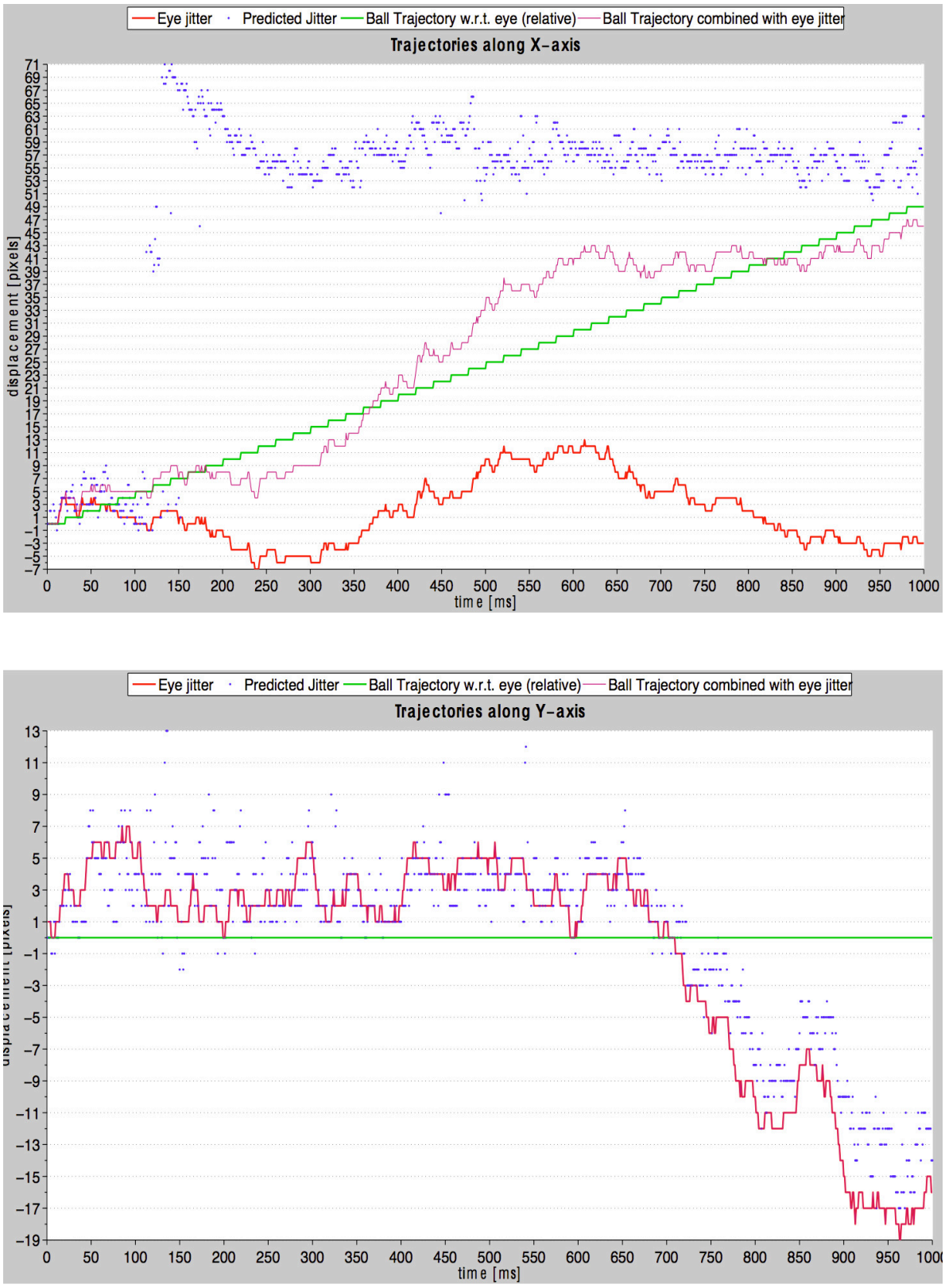


Figure 5.6: Inferred trajectories using OFF cell firing rates, after first run of FBD on artificial stimulus

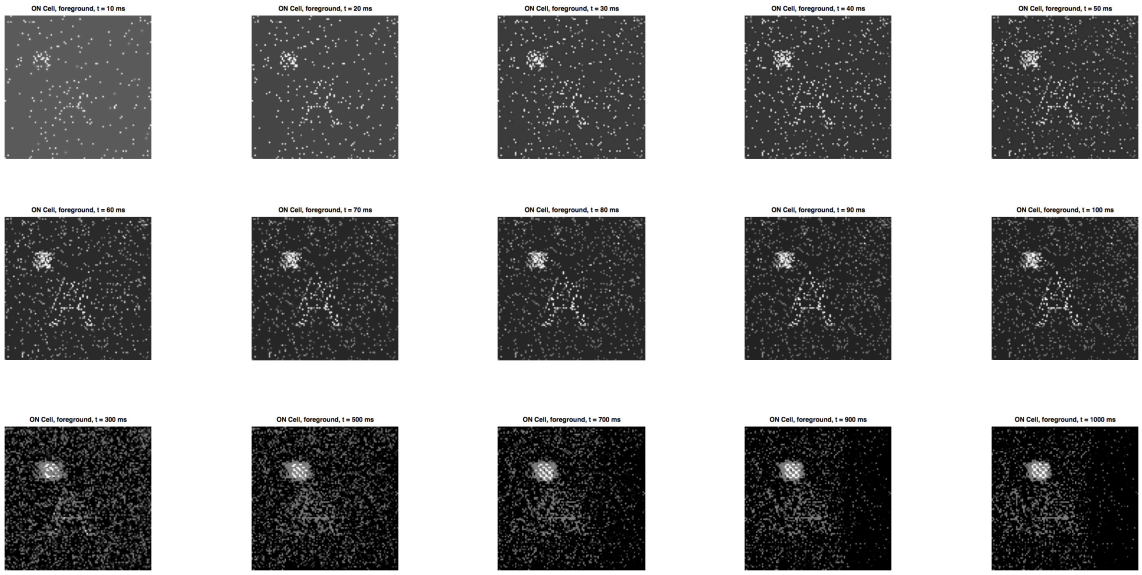


Figure 5.7: Inferred foreground grayscale image using ON cell firing rates, after second run of FBD on artificial stimulus

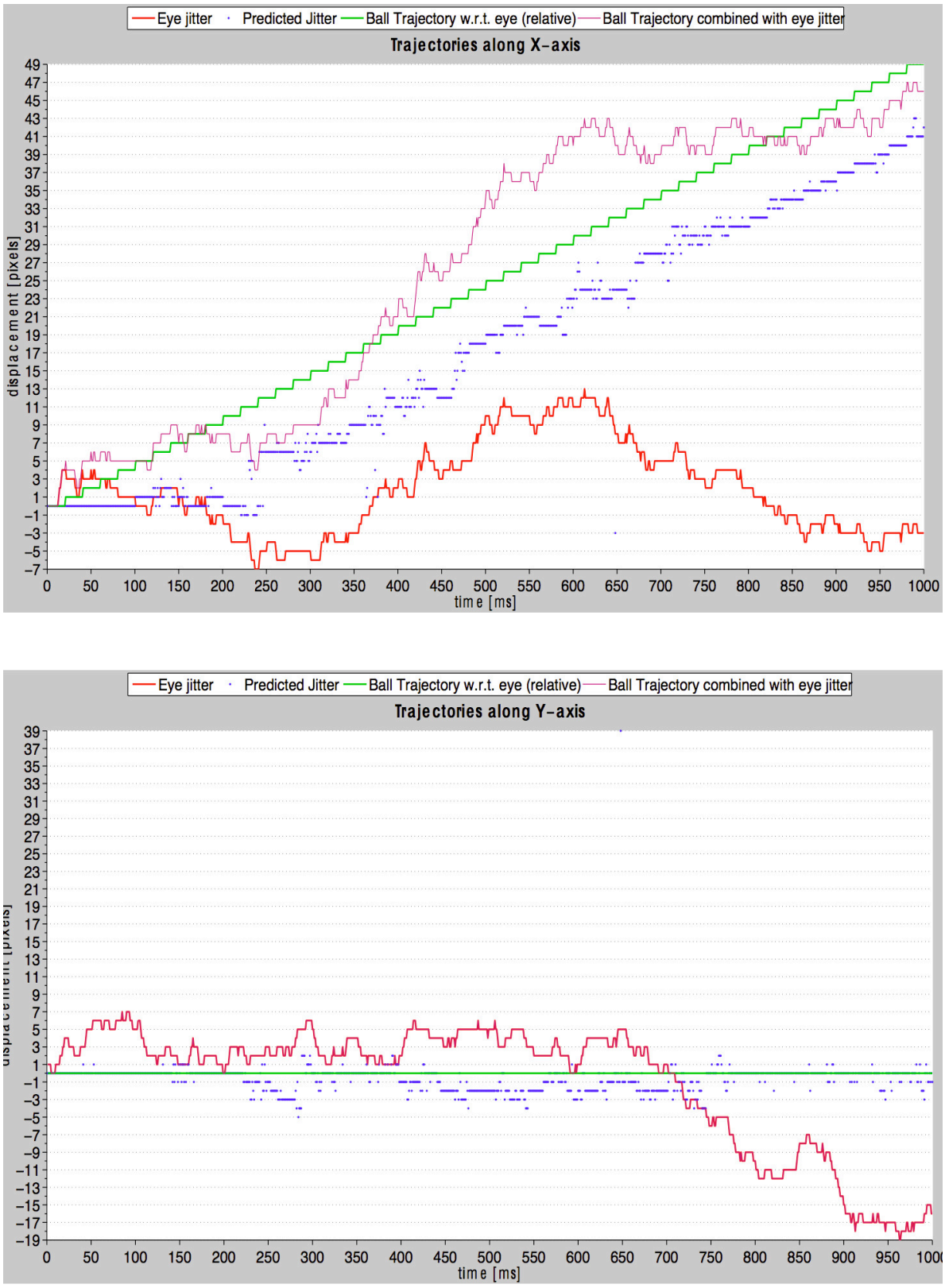


Figure 5.8: Inferred trajectories using ON cell firing rates, after second run of FBD on artificial stimulus

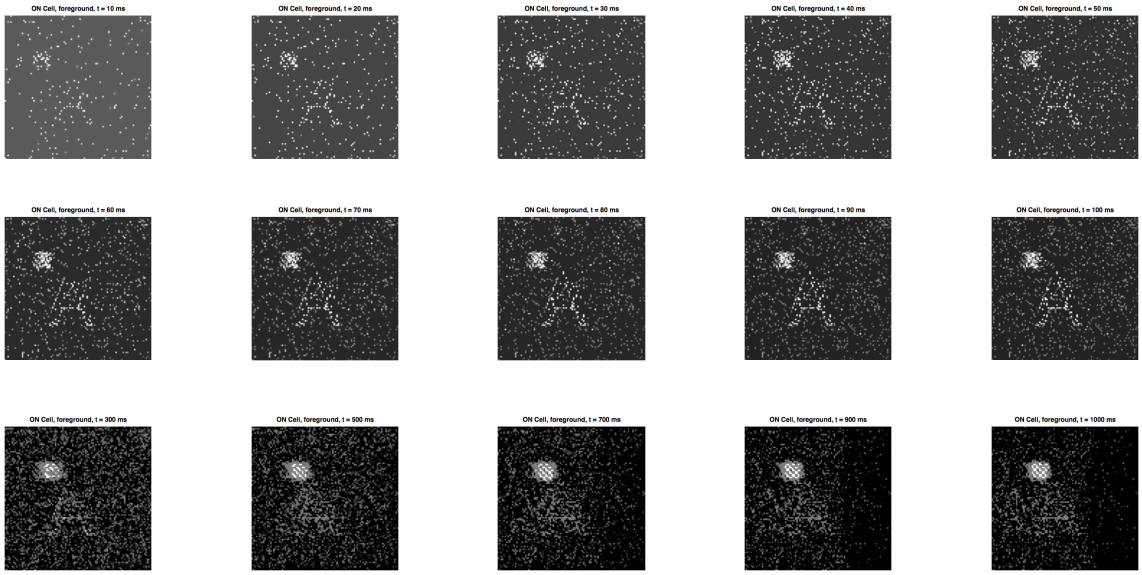


Figure 5.9: Inferred foreground grayscale image using OFF cell firing rates, after second run of FBD on artificial stimulus

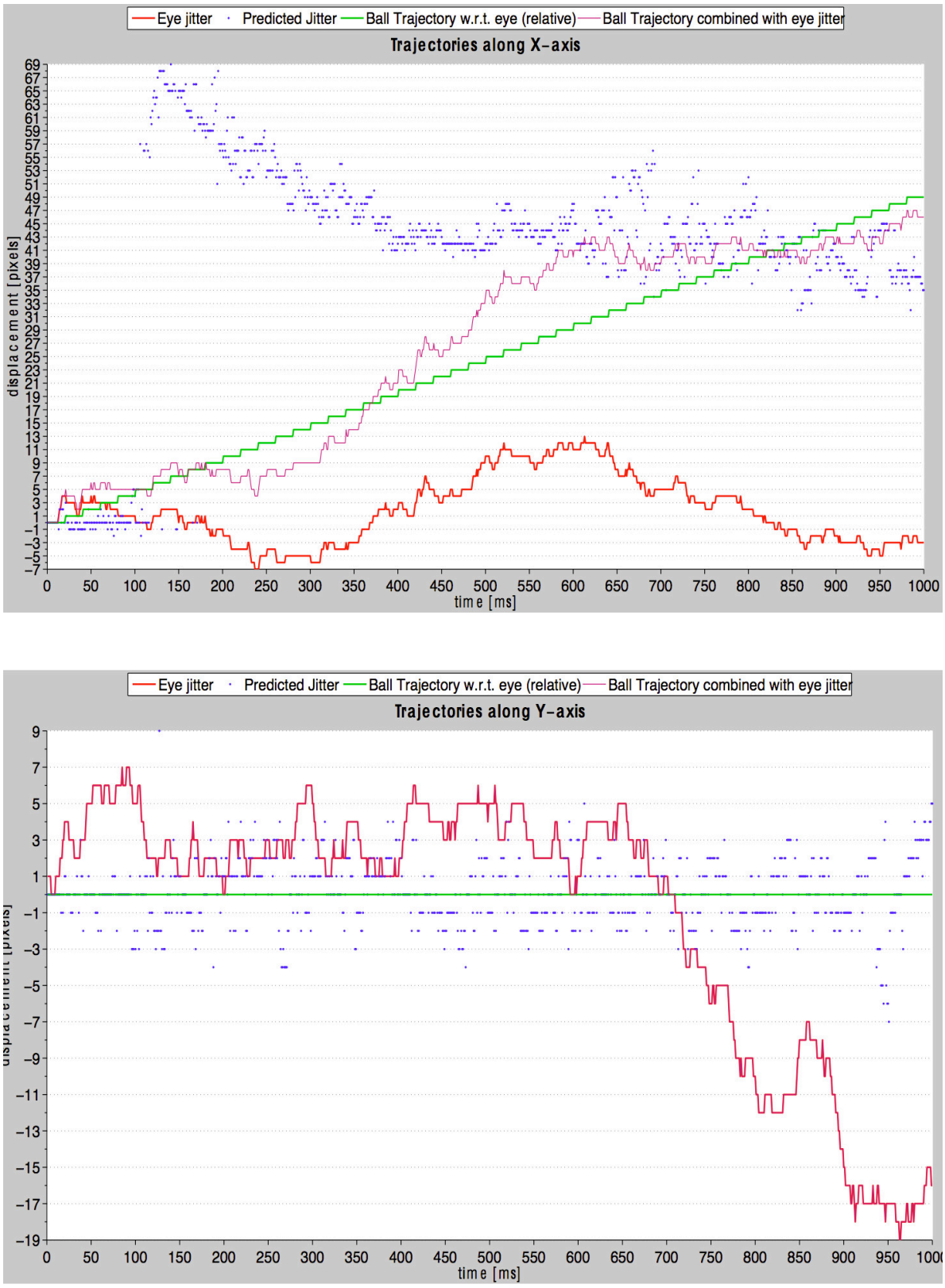


Figure 5.10: Inferred trajectories using OFF cell firing rates, after second run of FBD on artificial stimulus

5.2 Natural Stimulus in Grayscale

We consider a scene where an aeroplane is flying across the sky [3]. The scene consists of a white aeroplane with black wings and tail flying over gray clouds and the sky in Figure 5.11. The number at the bottom left and the text in the middle were present in the original video and were not discarded for our tests. We pre-process the video so that it has only 18 gray colors. A histogram of colors in the first frame is shown in Figure 5.12.

Figure 5.11: Aeroplane stimulus, first frame

The original video has 68 frames and plays at a frame rate of 15 frames per second. For our experiments, we made a video stimulus using the first 17 frames of the video, each frame lasting for 30 ms, so that the stimulus duration is 510 ms. Frames of the input stimulus are shown in Figure 5.13.

We set $\Delta t = 30$ frames and $bthresh = 0.1$ for the procedure for extracting object spikes. The results are shown in Figures 5.14, 5.16, 5.18, 5.20 respectively. Each figure shows the inferred gray image by FBD over time, sequentially.

As shown in Figure 5.14, using ON cell spike trains $\{g\_on_k(t)\}\,\forall k$ at every timestep t, we recover the light gray parts of the scene such as the clouds, the number at the bottom left, and the aeroplane's shape and trajectory. We don't recover the aeroplane entirely because it moves over static parts of the scene that have colors similar to it, such as the light grayish clouds. But since it flies mostly across the dark sky, a significant part of the aeroplane is visible. Also, one may

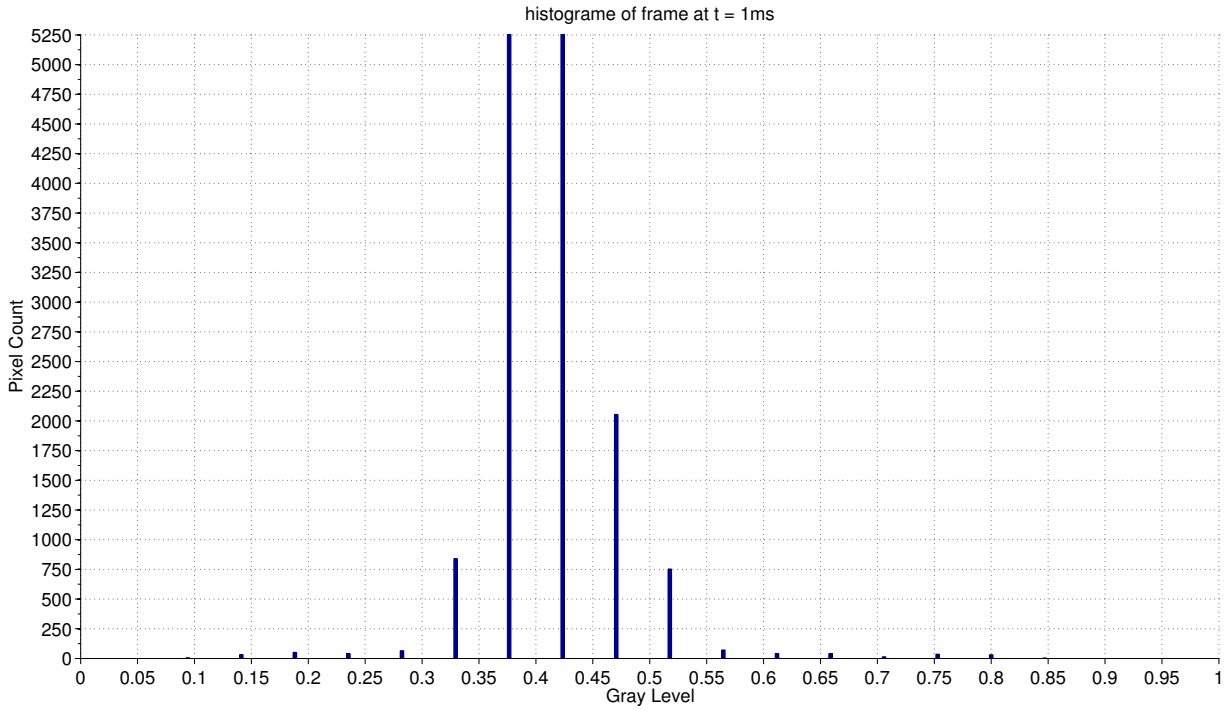


Figure 5.12: Aeroplane stimulus, first frame histogram

notice that the shape of the aeroplane area is longer that in the original input. This is because when the aeroplane moves from one place to another, smoothly, the spikes that neurons emitted at the prior location of the aeroplane still affect the belief of FBD that there is something white in that location. As the aeroplane moves towards left, the whiteness of the initial location of the plane decreases. As shown in Figure 5.15, FBD is able to recover the shape of the eye jitter trajectory, and is off the actual trajectory by 4 pixels in the horizontal direction. This is most likely the offset induced by the motion of the aeroplane over time.

With OFF cell spike trains, we have good success in recovering the clouds and their shapes, as shown in Figure 5.16. This is a good image of the background, and OFF cell spike trains are useful because the sky is mostly gray, and they are able to help recover the gradients across the sky and clouds effectively well. And as shown in Figure 5.17, FBD is able to recover the trajectory almost perfectly. The offset of 2 pixels over time in the horizontal direction can indicate a small motion of the clouds.

Given the background image from the ON cell spike train stimulus, running FBD again isn't useful because the image contains the aeroplane's shape and trajectory. Therefore, the results aren't meaningful as shown in Figure 5.18, and neither is the trajectory useful as shown in Figure 5.19.

The results of using the background from OFF cell neurons in Figure 5.20 can be explained as follows: We are able to recover the dark gray colored sky and gray clouds and gradients because the spike trains $\{g\_off_k(t)\}\,\forall k, t$ contain a high number of spikes for dark gray colors and low number of spikes for light gray colors. Therefore, the background already accounts for most of the spikes in the stimulus, and hence the object reconstruction system will believe that there are few spikes coming from the background, and the resulting figure will consist of light gray colors. Further, a dark gray patch is visible in the recovered foreground from the OFF cell neurons, which denotes the aeroplane patch moving because the aeroplane is white and the area from that patch will send a small number of spikes. We can conclude that even though the spikes from the aeroplane patch are few, there are significant enough so that we may distinguish between the sky and the background.

The trajectories in Figure 5.21 show that FBD does recover some motion of the aeroplane towards the left after 250 ms. There is no longer any eye jitter because we stabilize the stimulus using recovered jitter trajectory $x(t)\,\forall t$ from the first run of FBD.

We may conclude that the results of both ON and OFF cell stimuli must be combined to yield exactly what we see.

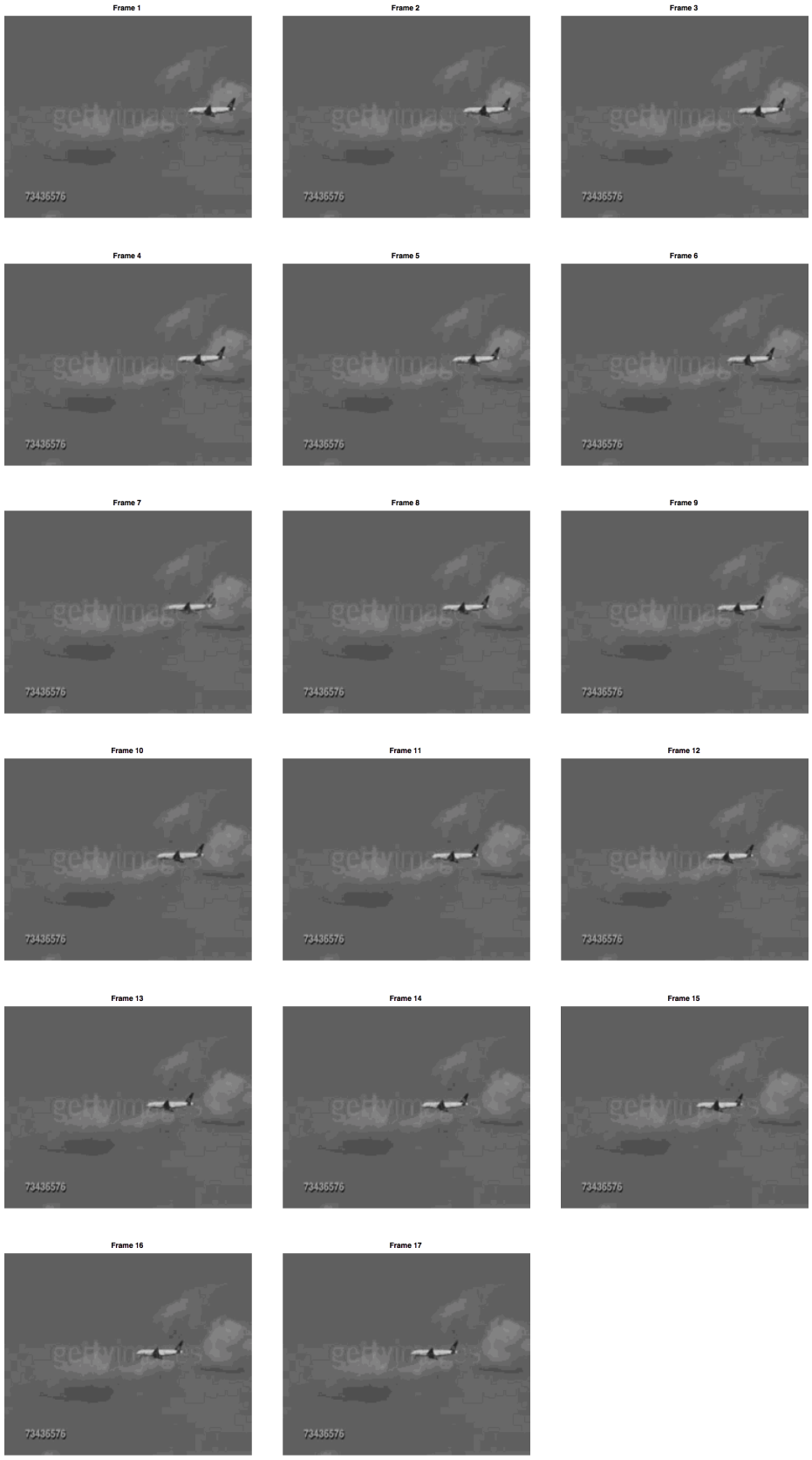


Figure 5.13: The first 17 frames of original video used to generate the stimulus as input to the object reconstruction system.

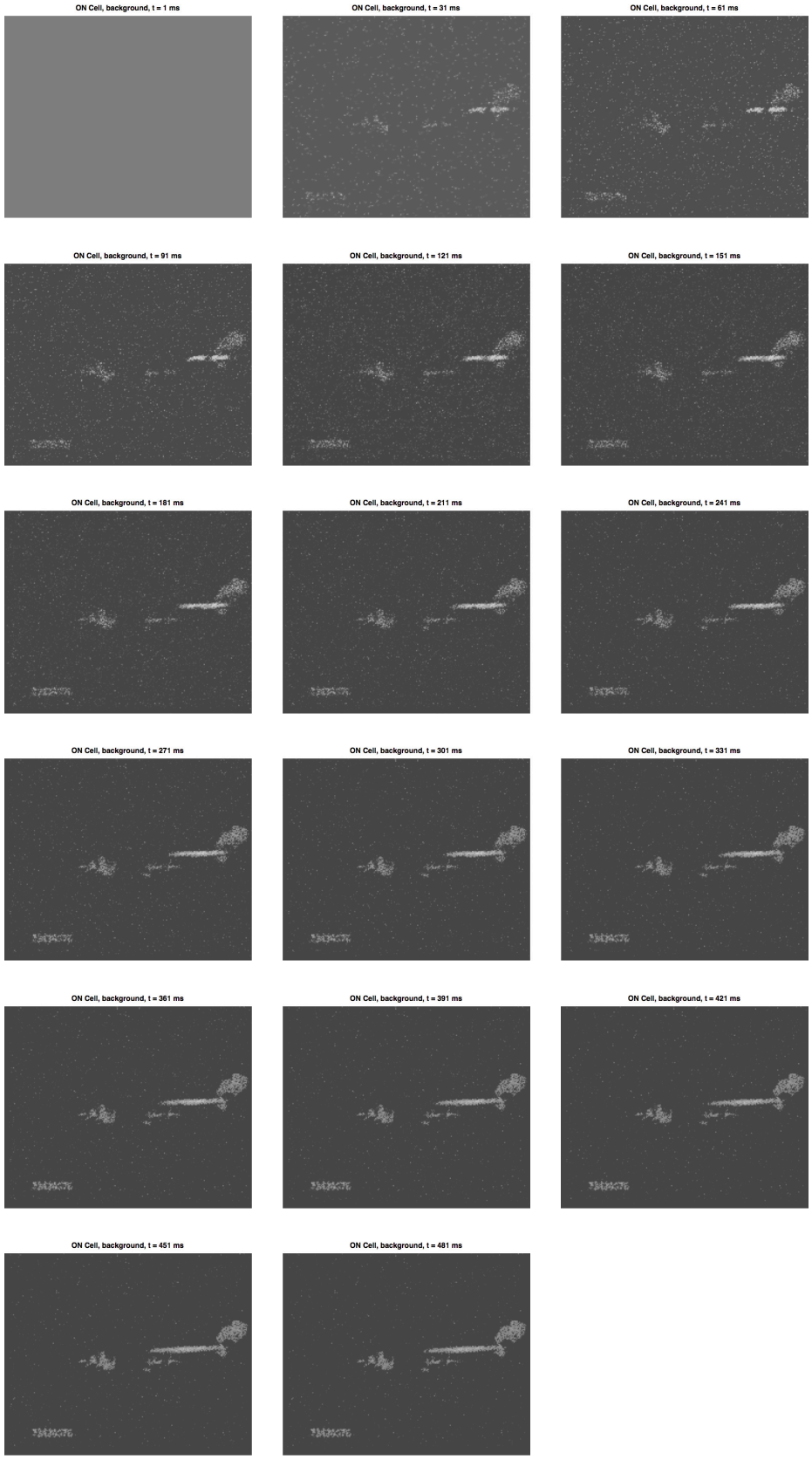


Figure 5.14: Inferred background grayscale image using ON cell firing rates, after first run of FBD on natural stimulus

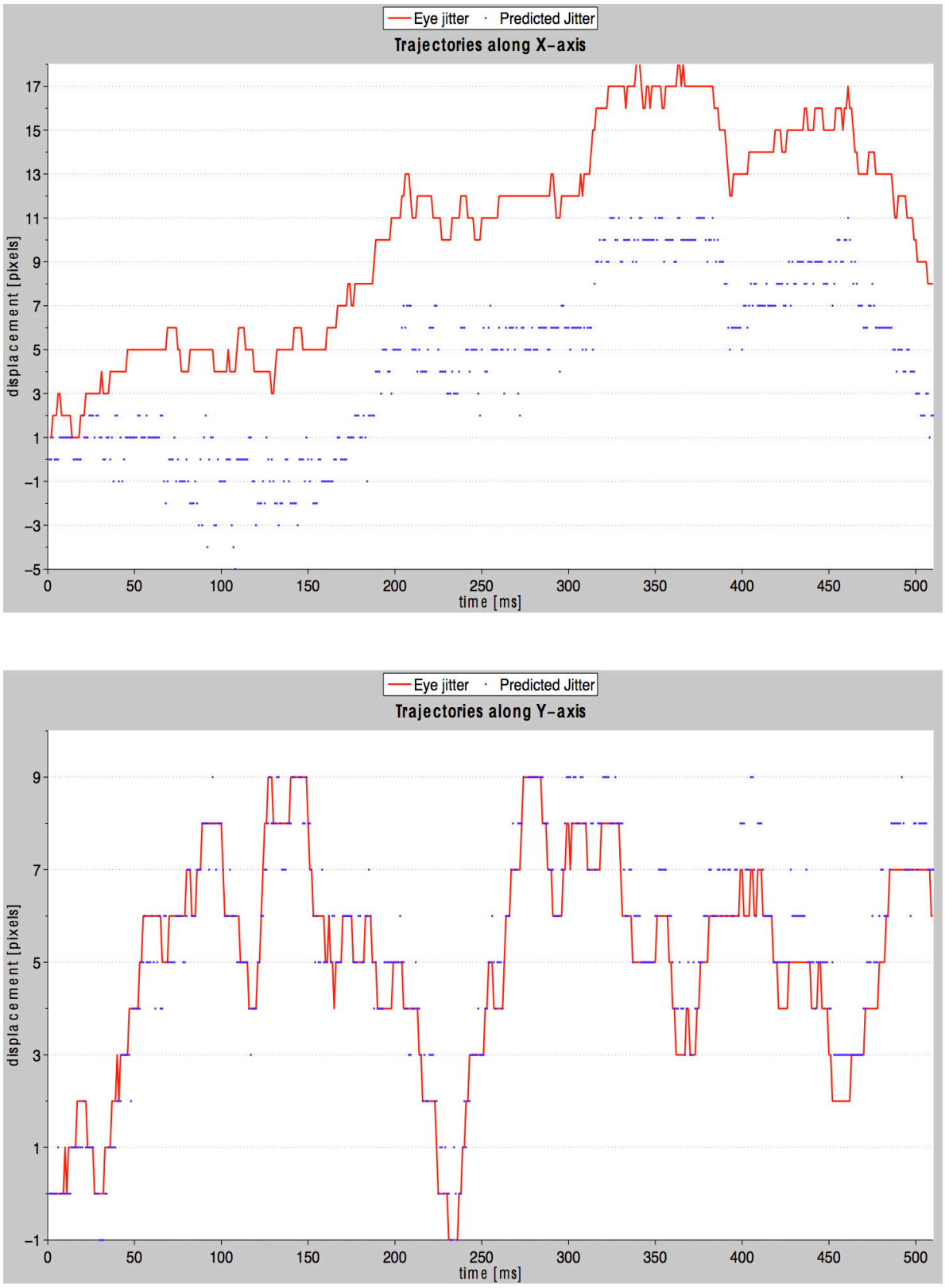


Figure 5.15: Inferred trajectories using ON cell firing rates, after first run of FBD on natural stimulus

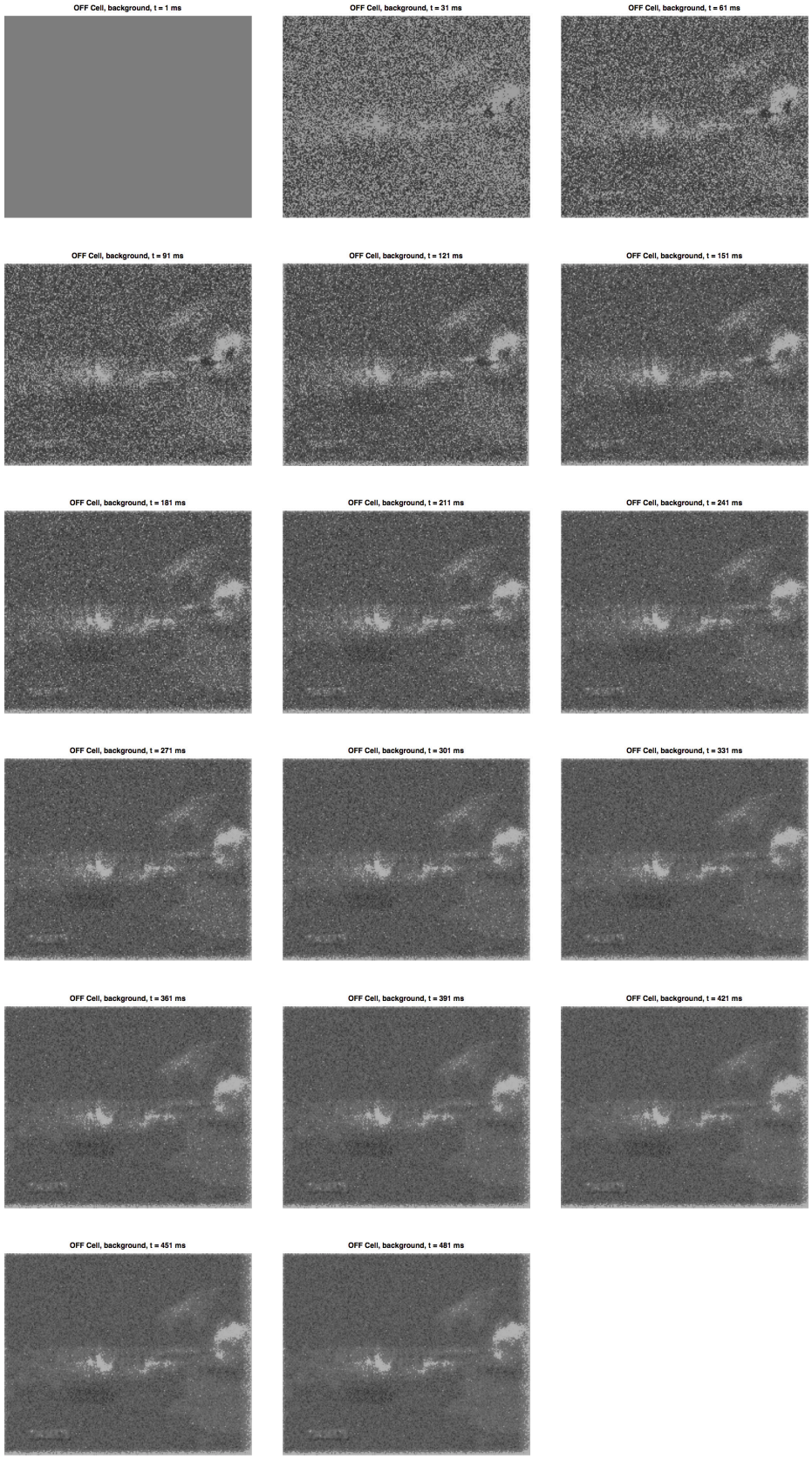


Figure 5.16: Inferred background grayscale image using OFF cell firing rates, after first run of FBD on natural stimulus

Figure 5.17: Inferred trajectories using OFF cell firing rates, after first run of FBD on natural stimulus

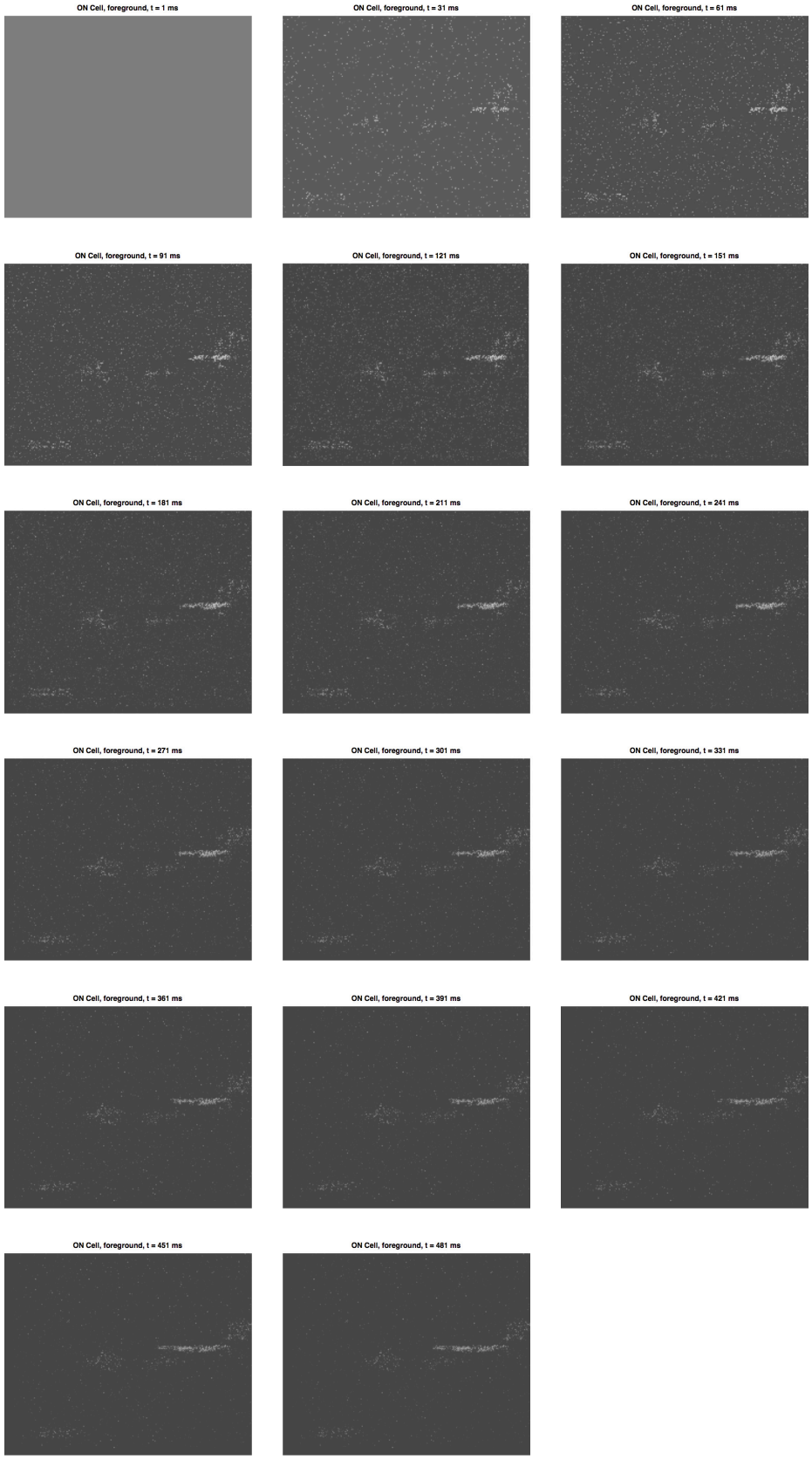


Figure 5.18: Inferred foreground grayscale image using ON cell firing rates, after second run of FBD on natural stimulus

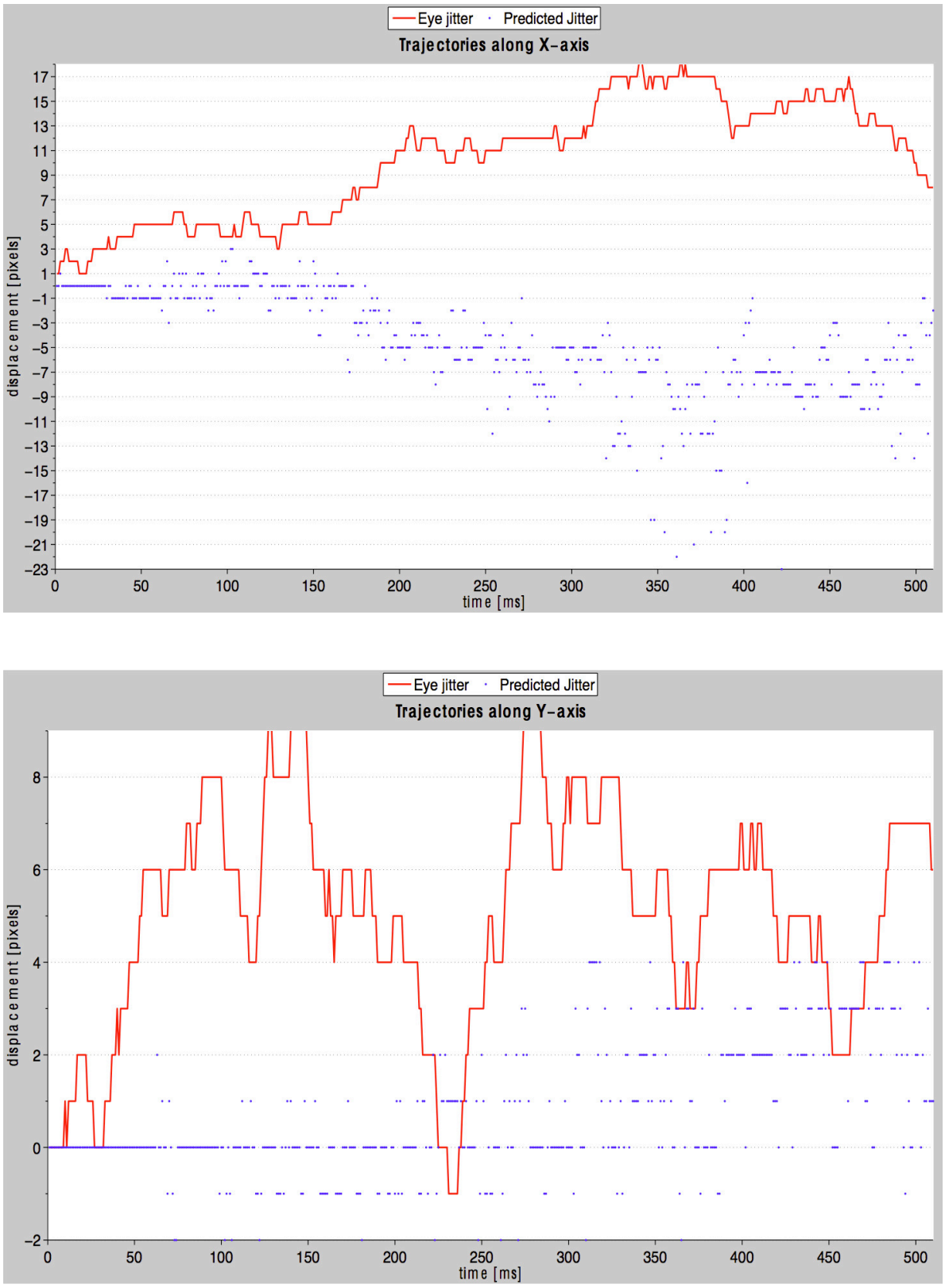


Figure 5.19: Inferred trajectories using ON cell firing rates, after second run of FBD on natural stimulus

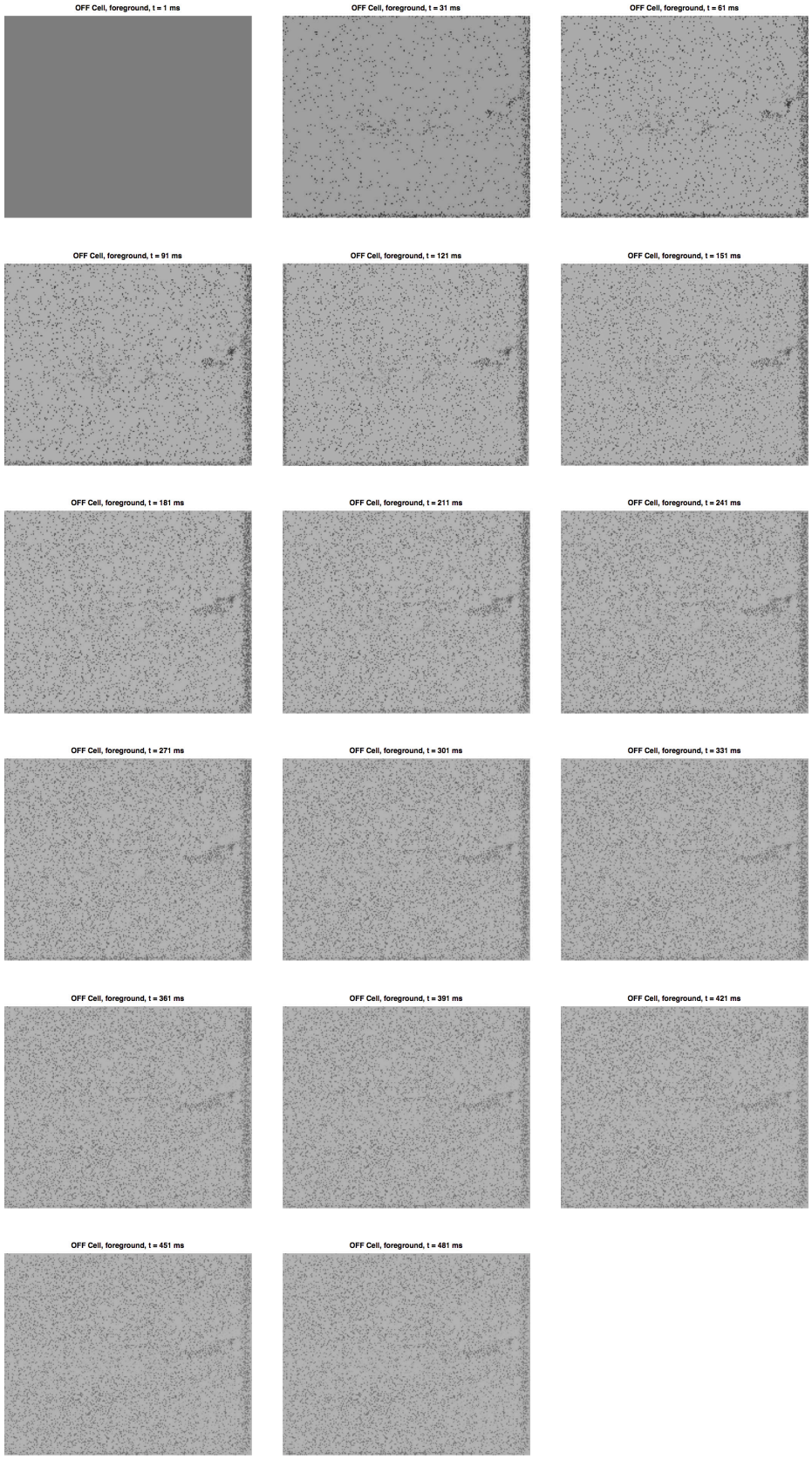


Figure 5.20: Inferred foreground grayscale image using OFF cell firing rates, after second run of FBD on natural stimulus

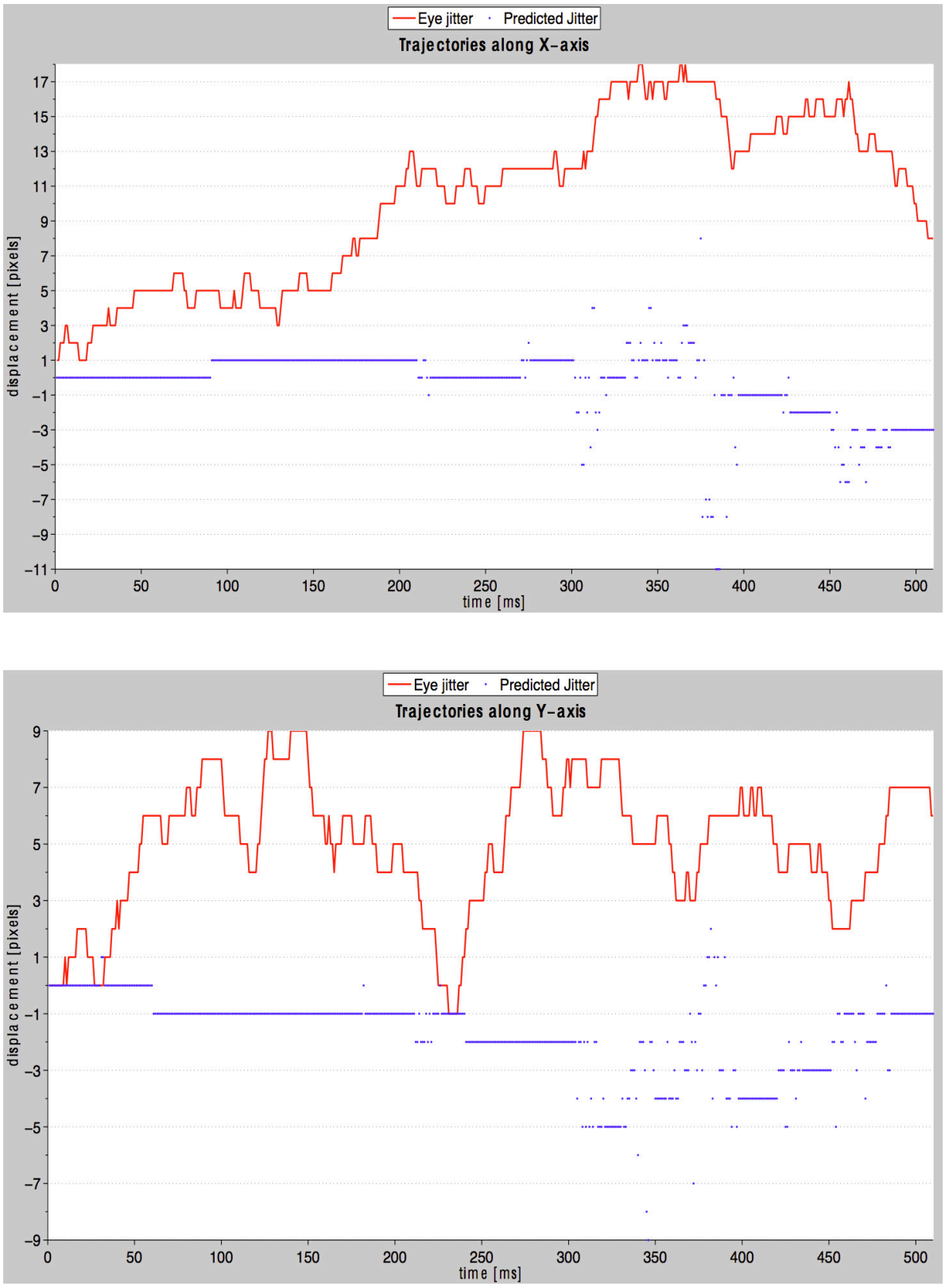


Figure 5.21: Inferred trajectories using OFF cell firing rates, after second run of FBD on natural stimulus

Chapter 6

Discussion

In this thesis, we discussed how the visual system can view static objects and dynamic objects in natural scenes using sparse and corrupted spikes from the retina using the Factorized Bayesian Decoder (FBD) proposed by Burak et.al. We also discussed how FBD can be implemented to view objects in grayscale images of natural scenes. We proposed a method to reconstruct the structure and details of an object that moves linearly across the retina, given the constraint that the neurons viewing the object emit fewer spikes at different locations on the retina, than for a static object.

The use of ON and OFF neuron stimuli as discussed in Chapter 5 seems necessary for dealing with natural grayscale stimuli, and conforms with the biological presence of such neurons in the retina. The spike trains from both kinds of neurons can help infer different parts of a scene, and the results can be combined to produce a coherent image of what we see.

A possible extension of this object reconstruction process is if we can use particle filters for tracking motion. Particle filters are used in computer vision for tasks like adaptive background subtraction, and thus can infer object motion for multiple objects in a scene. A particle filters tracks the distribution of color intensities at every pixel in the image, and infers from this distribution if at any time, the pixel is a part of the background or of a moving object. This is efficient and better than FBD as it doesn't track probabilities of every possible grayscale color at every pixel. However, particle filters work on pixel color values and not on spike trains. A particle

filter that can use spike train stimuli and extract foreground and background objects would be faster than a grayscale version of FBD, and efficient for dealing with scenes in grayscale.

One of area for improvement of our proposed solution is its inability to deal with multiple objects moving simultaneously in the same scene, such as a scene of linearly moving balls on a billiards table. And this is because FBD tracks global motion of the entire frame. When 2 or more balls in the scene move differently, FBD cannot track all objects simultaneously and may yield a poor result. This would require that we somehow mark the spikes from different objects as separate entities and track their motions separately. Baccus et.al. discovered the presence of a special kind of ganglion cell, called "Object Motion Sensitive" ganglion cell, or OMS cell in salamander retina [2]. This OMS cell is able to correctly distinguish object motion from retina jitter regardless of what objects are present in the scene. One can further investigate how information about object motion from such a cell can help in tracking multiple objects, and if this cell can help track object motion in scenes with non-linear trajectories of motion, such as that of a wheel with spokes spinning.

Further, it would be interesting to examine how we can make this object reconstruction technique more efficient. Presently, while implementing grayscale, FBD requires that we keep pixel probability maps for every possible grayscale value which is inefficient when there are so many gray values. How would this work for color images? Dealing with color is important and interesting, because it gives a better contrast in detail to objects and backgrounds than grayscale, and most people see in color.

In conclusion, a simple decoder such as FBD is useful in explaining how the brain can infer images of static and dynamic scenes using spike train input, for objects moving in a linear fashion. For explaining how the cortex can infer other kinds of motion, we may need to examine other statistical techniques such as particle filters or simulations of OMS cells.

Chapter 7

Bibliography

[1] Tiger grayscale image. http://cs.nyu.edu/ fergus/teaching/vision/tiger.jpg. Accessed: 2015-07-20. 3.2

[2] Stephen A. Baccus, Bence P. O'lveczky, Mihai Manu, and Markus Meister. A Retina Circuit that Computes Object Motion. *The Journal of Neuroscience*, pages 6807–6817, 2008. 6

[3] Arslan Basharat, Yun Zhai, and Mubarak Shah. Content Based Video Matching Using Spatiotemporal Volumes. *Journal of Computer Vision and Image Understanding (CVIU)*, 110: 360–377, 2008. Accessed: 2015-07-25. 5.2

[4] Yoram Burak, Uri Rokni, Markus Meister, and Haim Sompolinsky. Bayesian model of dynamic image stabilization in the visual system. *PNAS*, 2010. 1, 2, 2.2, 2.3

[5] Tim Gollisch and Markus Meister. Eye Smarter than Scientists Believed: Neural Computations in Circuits of the Retina. *Neuron*, 65:150–164, 2009. 1

List of Figures