A Plausible Neural Model for Perceptual waves: A report on "Dynamics of travelling waves in visual perception." - by Hugh R. Wilson, et. al.

Joel Gutierrez and Mark Valente

June 8, 2007

1 Introduction

This article gives a model for the visual phenomenon of travelling waves in perception. The basis of the model is the phenomenon of binocular rivalry, which is defined as "a phenomenon in which visual perception alternates between different images presented to each eye."[1] The reason this occurs is due to the connectivity and selectivity of neurons. Neurons can be connected to each other directly or indirectly, via some other neuron. There are also different types of neurons, some excite other neurons and others inhibit other neurons. Neurons are classified according to their morphology and behaviour. The attriubute of selectivity of a specific type of neuron is described as the neurons response to a particular stimulus. For example, some neurons might responed vigorouly when the eye is exposed to a black vertical line, while others respond weakly. A neurons response is characterized by the firing frequency of the neuron when it is exposed to a particular stimulus. The firing frequency is the rate at which the neuron sends messages to neighboring connected neurons. The basis of the model given by Wilson et. al. can be described by four neurons connected in a particular way, such that the increasing firing rate of one neuron causes the firing rate of the other neuron to decrease. Hence, there exists a rivalry between two neurons.

2 Description and Discussion

The behaviour of the firing rate of a single neuron, imagining that its firing rate is high at $t = 0$, can be modeled mathematically by the following simple firing rate model.

$$\tau \frac{dT}{dt} = -T$$

This states that the firing rate of a neuron will exponentially decrease over time, as long as the neuron receives no stimulus after $t = 0$. However, neurons are not isolated in our brains. Therefore, in order to include the contribution of neighboring neurons, the model must consider the influence of some stimuli. This can be accounted for by the Naka-Rushton fucntion, which is a Sigmoid.

$$f(s) = \begin{bmatrix} f_{max}\frac{s^n}{\theta^n + s^n} & s \geq 0 \\ 0 & s \leq 0 \end{bmatrix},$$

where f_{max} is the maximal firing rate of the neuron, s is the amplitude of the stimulus, and θ is the value of s at which the firing rate is $\frac{1}{2}f_{max}$. The question may arise as to how Naka and Rushton decided that the response of a neuron when introducing some stimulus can be respresented by a function of this form. Imagine taking an isolated neuron and exposing it to a constant stimulus for some time. The firing rate of the neuron will increase to some maximal level and then decrease to some steady state. The reason for the decrease in firing rate is do to adaptation, which very much like fatigue in the muscle. Then, record the firing rate of the neuron at its steady state. Repeat this procedure for several values of s and you will see that the fitted function is a sigmoid. Therefore, the mathematical model for a neuron exposed to some stimuli is

$$\tau\frac{dT}{dt} = -T + f(s).$$

In this model the input of a rivaling neuron will take away from the firing rate of the opposing neuron. This is called inhibition. In order to account for the inhibition, adaptation, and some outer stimuli, the simple firing rate model is for a neuron with outer stimuli, adaptation, and inhibition is expressed as

$$\begin{aligned}
\tau \frac{dT}{dt} &= -T + \frac{100P_T^2}{(10+H_T)^2 + P_T^2} \\
\tau_{I_T} \frac{dI_T}{dt} &= -I_T + T \\
\tau_{H_T} \frac{dH_T}{dt} &= -H_T + 2T \\
P_T &= max \begin{bmatrix} E_T \\ 0 \end{bmatrix},
\end{aligned}$$

where T is the firing rate of a neuron that is selective to target shaped stimuli, H_T is the adaptation variable of the neuron, I_T is the firing rate of the inhibitor neuron that is driven by the T neuron, P_T is the input into the T neuron, and E_T is some outer stimuli. The two rivaling neuron model the can then be expressed as

$$\begin{aligned}
\tau \frac{dT}{dt} &= -T + \frac{100P_T^2}{(10+H_T)^2 + P_T^2} \\
\tau_{I_T} \frac{dI_T}{dt} &= -I_T + T \\
\tau_{H_T} \frac{dH_T}{dt} &= -H_T + 2T \\
P_T &= max \begin{bmatrix} E_T - I_S \\ 0 \end{bmatrix}, \\
\tau \frac{dS}{dt} &= -S + \frac{100P_S^2}{(10+H_S)^2 + P_S^2} \\
\tau_{I_S} \frac{dI_S}{dt} &= -I_S + S \\
\tau_{H_S} \frac{dH_S}{dt} &= -H_S + 2 \\
P_S &= max \begin{bmatrix} E_S - I_T \\ 0 \end{bmatrix}.
\end{aligned}$$

Notice that the only difference here is that the input to each neuron accounts for the inhibition from their oppositional neighbor.

These equations decribe a very basic model for rivalry in a system of two neurons.

The two rivaling neuron model can be considered the basic unit of rivalry. Hence, a network of rivaling neurons will be composed of many units of rivaling neurons. It is in a network of neurons that waves can occur. First, the model for a network of neurons must incorporate the influence of all neurons in the network on any particular neuron within the network. To do this, Wilson et. al. proposes that the inhibition from opposing neurons be summed over and weighted. In order to see waves of dominance in the model inhibition by opposing neurons is all that is necessary[2]. The model proposed by Wilson et. al. also incorporates collinear facilitation, which is a positive contribution from neighboring cells of the same type.

The complete model presented by Wilson et. al. considers a network of neurons that are coupled to each other, and it is by this coupling that neighboring neurons can contribute to the input into one particular neuron. The model can be expressed mathematically as follows,

$$\begin{aligned}
\tau\frac{dT}{dt} &= -T + \frac{100P_T^2}{(10+H_T)^2+P_T^2} \\
\tau_{I_T}\frac{dI_T}{dt} &= -I_T + T \\
\tau_{H_T}\frac{dH_T}{dt} &= -H_T + 2T \\
P_T &= max\left[\begin{array}{c} E_T - 0.27\sum_k I_{S_k} exp\left(-\frac{x_{nk}^5}{\sigma^5}\right) + g\sum_{k\neq n} T_k exp\left(-\frac{x_{nk}^5}{(2\sigma)^5}\right) \\ 0 \end{array}\right], \\
\tau\frac{dS}{dt} &= -S + \frac{100P_S^2}{(10+H_S)^2+P_S^2} \\
\tau_{I_S}\frac{dI_S}{dt} &= -I_S + S \\
\tau_{H_S}\frac{dH_S}{dt} &= -H_S + 2 \\
P_S &= max\left[\begin{array}{c} E_S - 0.27\sum_k I_{T_k} exp\left(-\frac{x_{nk}^5}{\sigma^5}\right) + g\sum_{k\neq n} S_k exp\left(-\frac{x_{nk}^5}{(2\sigma)^5}\right) \\ 0 \end{array}\right].
\end{aligned}$$

In order for the model to reproduce experimental data the time constants were chosen to be:

$$\begin{aligned}
\tau &= 20ms \\
\tau_I &= 11ms \\
\tau_H &= 900ms.
\end{aligned}$$

We attempted to model the rivaling between two neurons. Therefore, we omitted summation over neighboring inhibition and we omitted collinear facilitation all together, since the sum is over $k \neq n$. In doing so, we used the following parameters for get oscillation in the firing rates of the rivaling neurons:

$$\begin{aligned} \tau &= 20ms \\ \tau_I &= 11ms \\ \tau_H &= 300ms. \end{aligned}$$

Also, we change the weight on the summation over the inhibition to 0.45 and the weight on T in the adaptation equation to 0.47, rather than 2.

It was our ambitious goal to have a 2-D simulation to show the occurance of waves. One thing to consider in the computation of a 2-D model is how to compute the distance between the n^{th} and k^{th} neuron. It is simply triangulation (see code).

Waves would occur in the lattice due to the fact that not all neurons of one type will switch stability at the same time. It would be one, or maybe a few, neurons of one type switching stability at one moment in time, and thereby contributing strongest to their nearest neighbor's excitation. Hence, creating a dominance wave propagating outward.

3 Conclusion

The model presented by Wilson et. al. is a simple and plausible neural model describing the phenomenon of travelling waves in visual perception. It considers a very basic model for rivalry between two neurons and expands into a network of rivaling neurons. It is only when multiple neurons are considered that waves can occur. Wilson et. al. also incorporates collinear facilitation, which is in agreement with experimental data[3,4]. Even though we were not able to create a 2-D simulation, we were able to see the bistable behaviour of the model. Hence, we could see how the model supports wave phenomena.

4 References

1. http://en.wikipedia.org/wiki/Binocular_rivalry

2. Rinzel, J., Terman, D., Wang, X. J. & Eremtrout, B. Propagating activity pattern in large scale inhibitory neuronal networks. *Science* **279**, 1351-1355 (1995).

3. Field, D. J., Hayes, A. & Hess, R. F. Contour integrations by the human visual system:evidence for a local 'association' field. *Vision Res.* **33,** 173-193 (1993).

4. Kamitani, Y. & Shimojo, S. Manifestation of scotomas created by transcranial magnetic stimulation of human visual cortex. *Nature Neurosci.* **2,** 767-771 (1999).

5 Code

This code models the dynamics between two rivaling neurons. It will produce a series of plots of the phase portrait with nullclines. These plots can then be made into a movie, which will show how the nullclines move as the neurons rival. I thought I had a movie, but the program I was using to put the movie together decided to fail today!

```
# Author : Joel Gutierrez
# Date   : 5/22/07
# Place  : UC Davis
# Class  : Chaos and Nonlinear Dynamics

# Project - Presentation of binocular rivalry model based on the following article.
# Dynamics of Travelling Waves in visual perception
# Hugh R. Wilson, Rondolph Blake, and Sang-Hun Lee

# Right now the system is ignoring the summations! This code will be used to make a
# movie of the phase portrait showing the nullclines moving in time, therefore
# demonstrating the bistability of a rivaling system.

# Import plotting routines
from pylab import *
from numpy import *

'''
To convert the rendered frames into a video use the command:
ffmpeg -i figs/null_%5d.PNG video/null.mpg


For better quality output:
ffmpeg -i figs/null_%5d.PNG -b 98000 video/null.avi


To view the video with mPlayer:
mplayer video/null.mpg
mplayer video/null.avi

To delete the frames:
rm figs/null_*.PNG

To delete the video:
rm video/null.mpg
'''
# Time
time_step = 0
```

```
time_interval = 500
dt = 0.25
time = []

# Control parameter for the model:
tau_E = 20
tau_I = 11
tau_H = 300
E_S = 10
E_T = 10
I_wt = 0.45
H_wt = 0.47

T = []
I_T = []
H_T = []
S = []
I_S = []
H_S = []


#T cell
T.append(20)
I_T.append(10)
H_T.append(15)

#S cell
S.append(10)
I_S.append(8)
H_S.append(3)

# Arrays used to calculate the nullclines.
Tarray = arange(-1,30,0.01)
Sarray = arange(-1,30,0.01)

# RKThreeD(Firing rate (T) time constant, Inhibition (I) time constant, Adaptation (H) ti
def RKThreeD( tau_E, tau_I, tau_H, p, E_fr, I_fr, H_fr, E_fcn, I_fcn, H_fcn, dt ):
        k1E = dt * E_fcn( tau_E, p, H_fr, E_fr )
        k1I = dt * I_fcn( tau_I,    I_fr, E_fr )
        k1H = dt * H_fcn( tau_H,    H_fr, E_fr )

        k2E = dt * E_fcn( tau_E, p, H_fr + k1E / 2.0, E_fr + k1E / 2.0 )
        k2I = dt * I_fcn( tau_I,    I_fr + k1I / 2.0, E_fr + k1I / 2.0 )
        k2H = dt * H_fcn( tau_H,    H_fr + k1H / 2.0, E_fr + k1H / 2.0 )

        k3E = dt * E_fcn( tau_E, p, H_fr + k2E / 2.0, E_fr + k2E / 2.0 )
```

```
        k3I = dt * I_fcn( tau_I,    I_fr + k2I / 2.0, E_fr + k2I / 2.0 )
        k3H = dt * H_fcn( tau_H,    H_fr + k2H / 2.0, E_fr + k2H / 2.0 )

        k4E = dt * E_fcn( tau_E, p, H_fr + k3E, E_fr + k3E )
        k4I = dt * I_fcn( tau_I,    I_fr + k3I, E_fr + k3I )
        k4H = dt * H_fcn( tau_H,    H_fr + k3H, E_fr + k3H )

        E_fr = E_fr  + ( k1E + 2.0 * k2E + 2.0 * k3E + k4E ) / 6.0
        I_fr = I_fr  + ( k1I + 2.0 * k2I + 2.0 * k3I + k4I ) / 6.0
        H_fr = H_fr  + ( k1H + 2.0 * k2H + 2.0 * k3H + k4H ) / 6.0

        return E_fr,I_fr,H_fr

# Differential equations describing the system
def T_dot(tau_T,P_Tplus,H_T,T):
    return (1.0/tau_T) * (-T + 100*P_Tplus**2/((10+H_T)**2 + P_Tplus**2))

def H_Tdot(tau_H,H_T,T):
    return (1.0/tau_H) * (H_wt*T - H_T)

def I_Tdot(tau_I,I_T,T):
    return (1.0/tau_I) * (T - I_T)

def S_dot(tau_S,P_Splus,H_S,S):
    return (1.0/tau_S) * (-S + 100*P_Splus**2/((10+H_S)**2 + P_Splus**2))
def H_Sdot(tau_H,H_S,S):
    return (1.0/tau_H) * (H_wt*S - H_S)

def I_Sdot(tau_I,I_S,S):
    return (1.0/tau_I) * (S - I_S)

# Main program solves the equations at each time step, which describe the state
# of two rivaling neurons.
for t in arange(0,time_interval,dt):
        #Store time value
        time.append(t)

        #Solve for input
        P_T = E_T - I_wt*I_S[time_step]
        P_S = E_S - I_wt*I_T[time_step]

        #Ensure input is non-negative
        P_Tplus = max(0,P_T)
        P_Splus = max(0,P_S)
```

```
#Solve the three equations governing the T cell and store values
T_firing, T_inhib, T_adapt = RKThreeD(tau_E, tau_I, tau_H, P_Tplus, T[time_step]
T.append(T_firing)
I_T.append(T_inhib)
H_T.append(T_adapt)

#Solve the three equations governing the S cell and store values
S_firing, S_inhib, S_adapt = RKThreeD(tau_E, tau_I, tau_H, P_Splus, S[time_step]
S.append(S_firing)
I_S.append(S_inhib)
H_S.append(S_adapt)

#Solve for the nullclines and store values
Tnull = []
Snull = []
#Calcuate and save the 100th nullcline
if time_step%100 == 0:
   for i in Tarray:
       p_t = max(0,E_T - I_wt*i)
       p_s = max(0,E_S - I_wt*i)
       Tnull.append(100*p_t**2/((10 + H_T[time_step])**2 + p_t**2))
       Snull.append(100*p_s**2/((10 + H_S[time_step])**2 + p_s**2))
   clf
   figure()
   title('H_T = ' +str(H_T[time_step])+ ', H_S = ' +str(H_S[time_step]))
   xlabel('Snull')
   ylabel('Tnull')
   plot(Sarray,Tnull,'r',Snull,Tarray,'b')
   axis([-1.0,30,-1.0,30])
   savefig('figs/null_'+str(time_step).zfill(5)+'.PNG')

#Increment the time index
time_step = time_step + 1
```