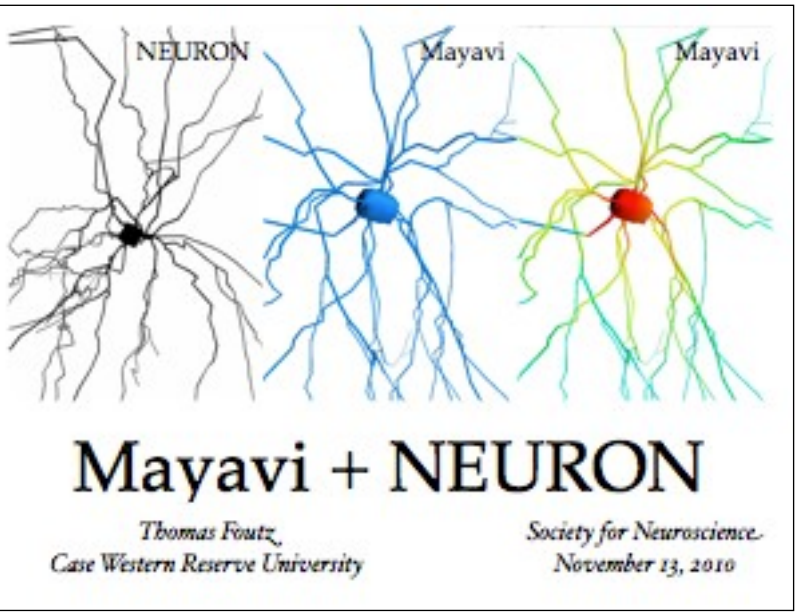


Mayavi + NEURON

Thomas Foutz
Case Western Reserve University

Society for Neuroscience
November 13, 2010

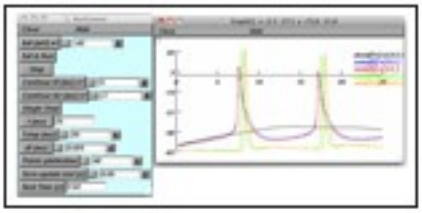
1

NEURON + InterViews

- Purpose:
  - *"... building and using computational models of neurons and networks of neurons."*
- Plots: InterViews
  - Stanford 1987-1991
  - NEURON 1991-Present

2

NEURON + InterViews



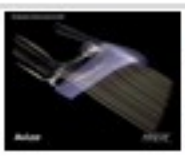
+

3D

?

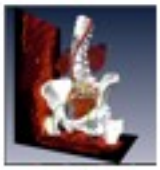
3

3D Visualization

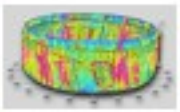


Avizo

> 5000 $
Matlab interface
LabView interface



amira

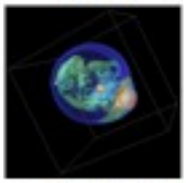


ScienceGL

SCIRun

Free/Open-source
Import Matlab Matrices

VTK



Free/Open-Source
Python Interface:
Mayavi

4

Enthought Python

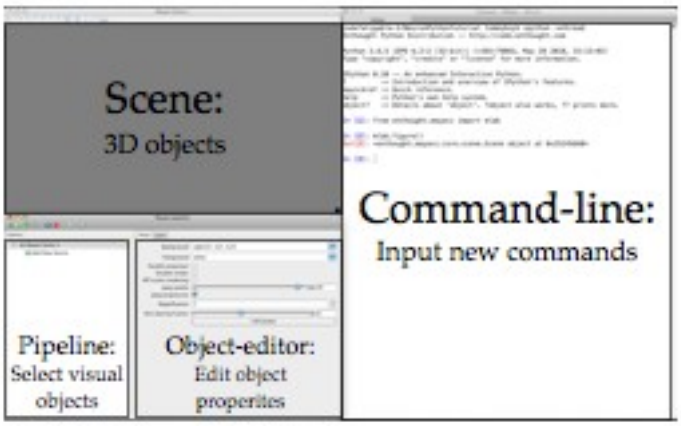
"Provides scientists with a comprehensive set of tools to perform rigorous data analysis and visualization."

- ***Pylab + Matplotlib + IPython***
  - Open source, object-oriented, python based MATLAB
  - Customizable 2D graphs
- ***Mayavi:*** 3d graphs
  1. SAGE: no-hastle use
  2. Local: *Interactive*, customizable

http://www.enthought.com/products/getepd.php

5

Mayavi



6

Mayavi: mlab

mlab: a simple scripting interface to Mayavi2 for 3D plotting.

- Mayavi ~ NEURON graphical user interface (GUI)
- mlab ~ HOC

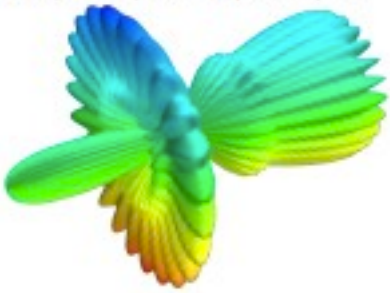
```
from enthought.mayavi import mlab
```

7

A Mayavi Example

```
# Create the data.
from numpy import pi, sin, cos, mgrid
dphi, dtheta = pi/250.0, pi/250.0
[phi,theta] = mgrid[0:pi+dphi*1.5:dphi,0:2*pi+dtheta*1.5:dtheta]
m0 = 4; m1 = 3; m2 = 2; m3 = 3; m4 = 6; m5 = 2; m6 = 6; m7 = 4;
r = sin(m0*phi)**m1 + cos(m2*phi)**m3 + sin(m4*theta)**m5 + cos(m6*theta)**m7
x = r*sin(phi)*cos(theta)
y = r*cos(phi)
z = r*sin(phi)*sin(theta)

# View it.
from enthought.mayavi import mlab
s = mlab.mesh(x, y, z)
mlab.show()
```



1_Spherical_Harmonics.py

8

Mayavi Recording

```
# Mesh was created:
s = mlab.mesh(x, y, z)
```

```
# Change Opacity:
s.actor.property.opacity = 1.0
```

9

Neuron3D

- **Purpose:**
  - Integrate Mayavi with NEURON
- *Open source, contributors welcome*
  - Clone repository:

```
git clone https://github.com/ttfoute99/Neuron3D.git
```

  - Import Code:

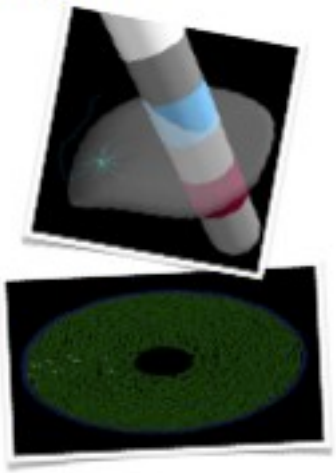
```
1 from neuron3d import *
```

https://github.com/ttfoute99/Neuron3D

10

Demonstration Projects

1. ***Deep Brain Stimulation (DBS)***
   - Subthalamic nucleus
   - Subthalamic projection neuron
     - Visualize Δ membrane voltage
   - DBS Electrode
2. ***Olfactory bulb network model***
   - Visualize spike data for granular and mitral neurons
   - Olfactory bulb cluster formation (Migliore et al. 2010)

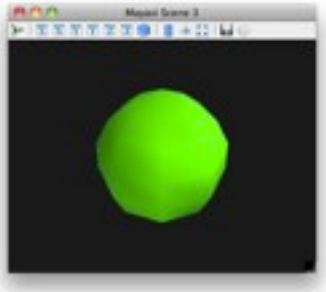


11

Basic Shapes

12

Single Sphere



```
x = [0]
y = [0]
z = [0]

mlab.points3d(x, y, z,
              color = (0.3, 1.0, 0.3),
              name = "sphere")
```

13

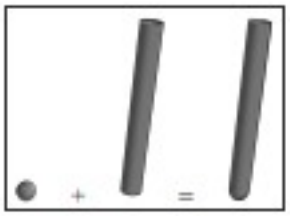
Electrode

- Setup:

```
from enthought.mayavi.mlab import *
fig = figure(bgcolor=(0.1, 0.1, 0.1))
```

- Sphere + Tube

```
# Sphere
points3d([0],
         [0],
         [0],
         scale_factor=1270,
         color=(0.5, 0.5, 0.5),
         name='Electrode Tip')
# Tube
plot3d([0,0],[0,0],[0,10000],
            color=(0.5, 0.5, 0.5),
            tube_radius= 1270 / 2,
            tube_sides=40,
            name='Electrode')
```



14

Electrode: Neuron3D

- Setup:

```
from enthought.mayavi.mlab import *
from neuron3D import *
fig = figure(bgcolor=(0.1, 0.1, 0.1))
```

- Electrode (from neuron3d):

```
# Instantiate Electrode Class
electrode = Electrode()

# Call display method
electrode.display()
```

- Finer resolution
- Capped
- Easier to manipulate trajectory

15

Project:
Deep Brain Stimulation

16

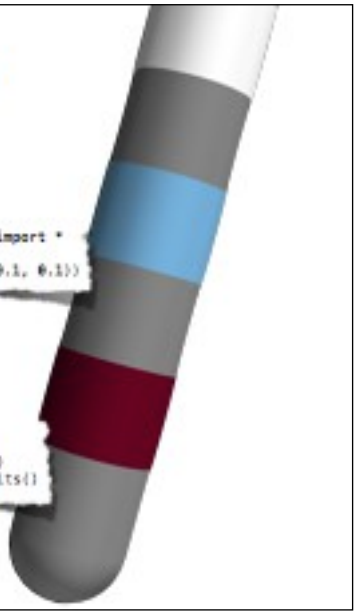
DBS Electrode

- Setup:

```
from enthought.mayavi.mlab import *
from neuron3D import *
fig = figure(bgcolor=(0.1, 0.1, 0.1))
```

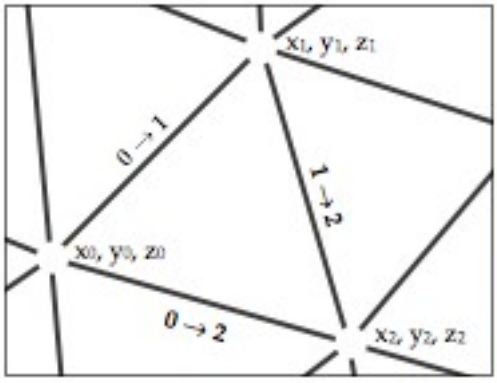
- DBS Electrode:
  (from neuron3d)

```
dbs = DBS()
dbs.display()
dbs.edit_traits()
```



17

Triangular Mesh



- Points
  - x_1, y_1, z_1
  - x_2, y_2, z_2
  - x_0, y_0, z_0
- Connections
  - $0 \rightarrow 1$
  - $0 \rightarrow 2$
  - $1 \rightarrow 2$

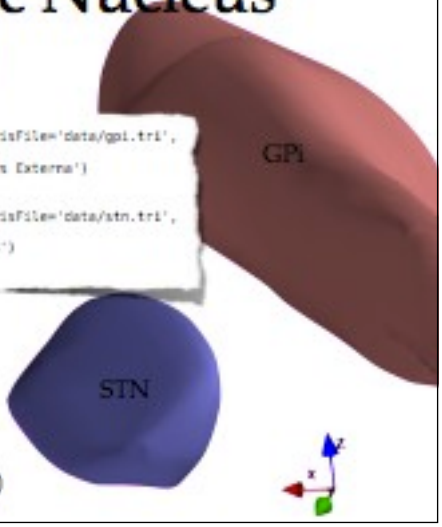
18

Subthalamic Nucleus

- STN, GPi and GPe Nuclei:

```
gpi = Nucleus(ptsFile='data/gpi.pts',trisFile='data/gpi.tri',
              color=(0.7, 0.4, 0.4),
              name='Globus Pallidus Pars Externa')
gpi.display()

stn = Nucleus(ptsFile='data/stn.pts',trisFile='data/stn.tri',
              color=(0.4, 0.4, 0.7),
              name='Subthalamic Nucleus')
stn.display()
```

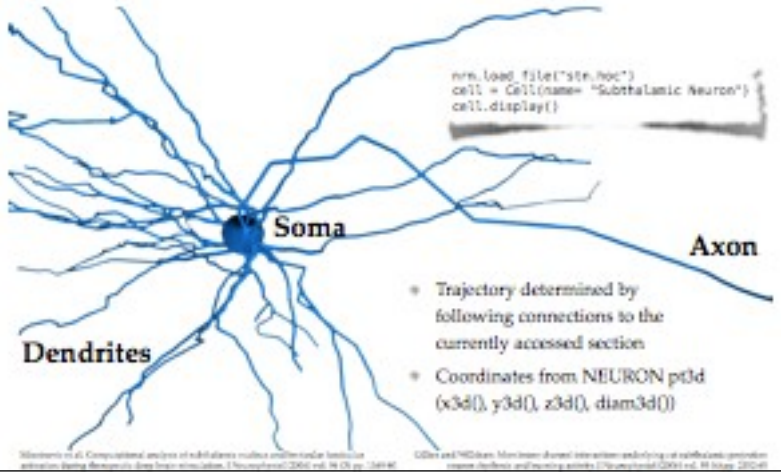


(Nuclei from Rhesus Macaque)

19

Subthalamic projection neuron

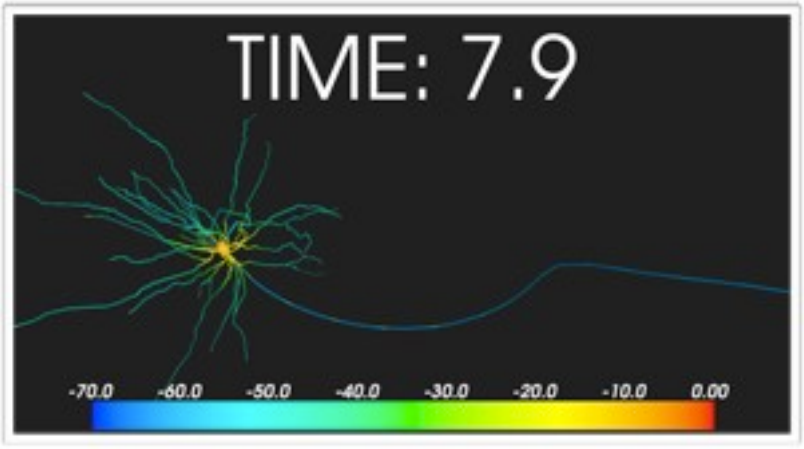
```
nrn.load_file("stn.hoc")
cell = Cell(name= "Subthalamic Neuron")
cell.display()
```



- Trajectory determined by following connections to the currently accessed section
- Coordinates from NEURON pt3d (x3d(), y3d(), z3d(), diam3d())

20

Subthalamic projection neuron



21

Deep Brain Stimulation

- Electrode

```
dbs = DBS()
dbs.display()
```

- Subthalamic nucleus

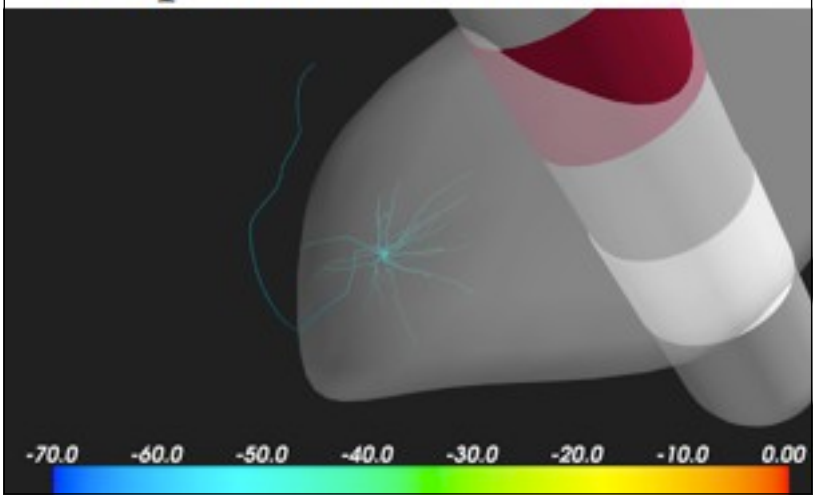
```
stn = Nucleus(ptsFile='data/stn.pts',
              trisFile='data/stn.tri',
              name='Subthalamic Nucleus',
              color= (0.75, 0.75, 0.75))
stn.display()
stn.set_opacity(0.5)
```

- Subthalamic projection neuron

```
nrn.load_file("stn.hoc")
cell = Cell(name='stn')
cell.display()
cell.move([0,0,-2000])
```

22

Deep Brain Stimulation

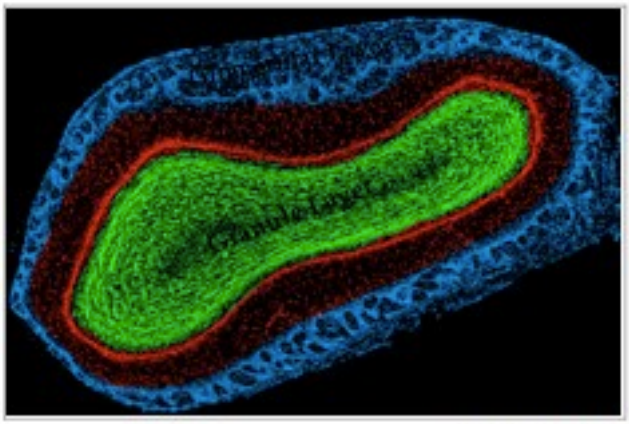


23

Project:
Olfactory bulb network

24

Olfactory Bulb

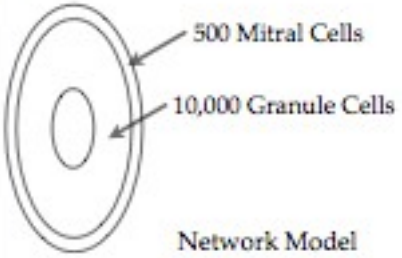


Matt Valley 2006. Public Domain.

25

Olfactory Bulb

Time	ID
65.800	250
65.800	251
65.800	252
65.800	253
101.95	250
101.95	251
101.95	252
101.95	253
128.20	250
128.20	251
128.20	252
128.20	253
144.65	450
144.65	451
144.65	452
144.65	453



Network Model

```
network = Network('data/somes-large-283-10s-20pc-g3.spk')
network.display()
network.play('Network')
```

Migliore and Shepherd. Dendritic action potentials connect distributed dendrodendritic microcircuits. J Comput Neurosci (2008) vol. 24 (2) pp. 207-21

26

Olfactory Bulb

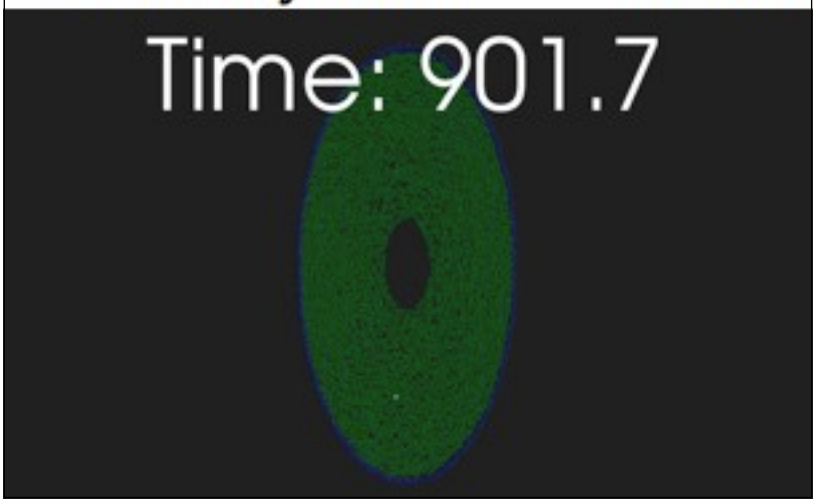
Flattened representation of network

Migliore and Shepherd. Dendritic action potentials connect distributed dendrodendritic microcircuits. J Comput Neurosci (2008) vol. 24 (2) pp. 207-21

27

Olfactory Bulb

900-1000 ms (?)



28

Summary

- Created visualizations for both multi-compartment and network simulations
- Successfully integrated Mayavi and NEURON
- Demonstrated rapid development of beautiful, interactive visualizations with off-the-shelf libraries

29

Neuron3D: Future Directions

- More efficient
- Rich GUI environment
- Infrastructure for network visualizations
- Download Neuron3d:
  - **https://github.com/tfoutz99/Neuron3D**
- Suggest, report, & contribute!

30

Questions?

31

Resources

- mlab scripting cookbook:
  - http://code.enthought.com/projects/mayavi/docs/development/html/mayavi/mlab.html
- Mayavi user manual:
  - http://code.enthought.com/projects/mayavi/docs/development/html/mayavi/
- Represent additional scalars on surfaces:
  - http://gael-varoquaux.info/blog/?p=128

mayavi-users@lists.sourceforge.net
tom.fouts@gmail.com

32