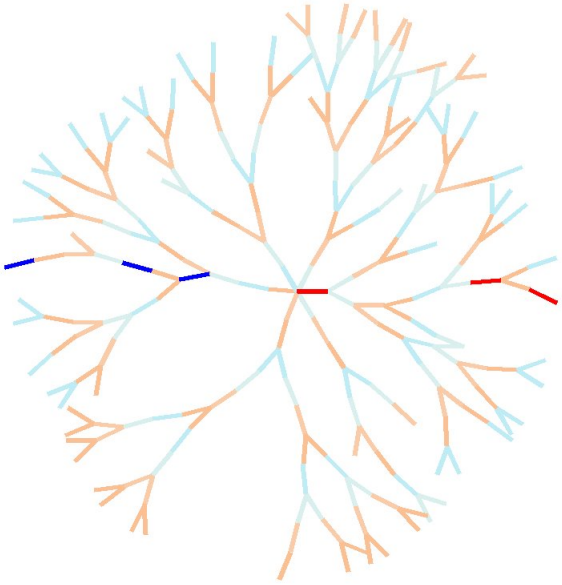

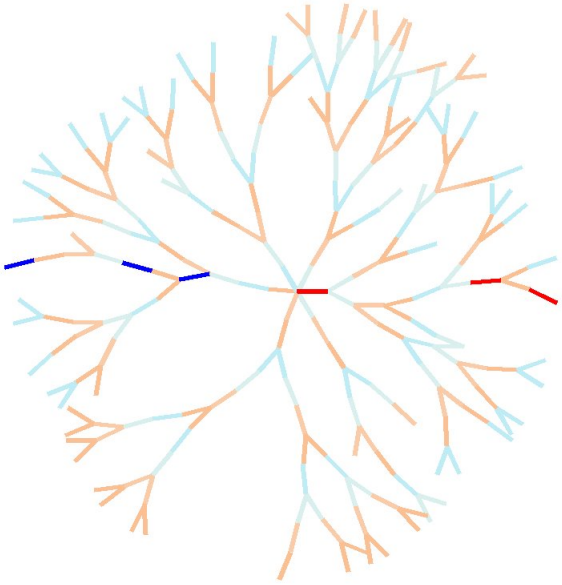

Parameters

```
# General retina
retina_width        = 400 * UM_TO_M
retina_height       = 400 * UM_TO_M
retina_grid_size    = 1 * UM_TO_M
retina_timestep     = 10 * MS_TO_S

# Cone Layer
cone_distance   = 10 * UM_TO_M
cone_density    = 100.0
cone_input_size = 10 * UM_TO_M

# Horizontal Layer
horizontal_input_strength   = 0.25
hoirzontal_decay_rate       = 0.01
horizontal_diffusion_radius = 1 * UM_TO_M

# Bipolar layer
bipolar_distance        = 10 * UM_TO_M
bipolar_density         = 100.0
bipolar_input_radius    = 10 * UM_TO_M
bipolar_output_radius   = 10 * UM_TO_M
```

```
# Build the starburst layer
starburst_distance  = 50 * UM_TO_M
starburst_density   = 1000.0
average_wirelength  = 150 * UM_TO_M
step_size           = 15 * UM_TO_M
decay_rate          = [0.1, 0.2, 0.3]
input_strength      = 0.5
diffusion           = ("Flat", [30 * UM_TO_M / retina_grid_size
'print_stop'



# Put parameters into lists
retina_parameters = [retina_width, retina_height, retina_gr
cone_parameters = [cone_distance, cone_density, cone_i
horizontal_parameters = [horizontal_input_strength, hoirzo
bipolar_parameters = [bipolar_distance, bipolar_density, b
starburst_parameters = [starburst_distance, starburst_den
runtime_starburst_parameters = [input_strength, decay_ra

# Set some default values in starburst parameters for the r
for parameter in runtime_starburst_parameters:
    if isinstance(parameter, (list, np.ndarray)):
        starburst_parameters.append(parameter[0])
    else:
        starburst_parameters.append(parameter)

'print_start'

# Bar paramters
framerate               = 60.0
movie_width             = 400
```

```
movie_height            = 400
bar_width               = 50.0  # Pixels (width = size in directio
bar_height              = 400
bar_speed               = 2000.0
bar_movement_distance   = 400.0
pixel_size_in_rgu       = 1.0    # rgu
```