Py4ET 2013: Creating Fixation Density Plot using Heat Maps

Setup Code

- **Load necessary Python Modules:**

In [78]:
```
import numpy as np
import matplotlib.cm as cm
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import brewer2mpl
```

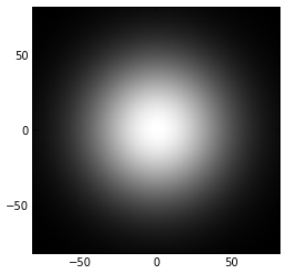
- **Switch working directory:**

In [79]:
```
cd D:\Dropbox\WinPython-32bit-2.7.5.0\my-code\ipython-notebooks
```

```
D:\Dropbox\WinPython-32bit-2.7.5.0\my-code\ipython-notebooks
```

Create 2D Gaussian Mask Template

In [80]:
```
# Gaussian Mask for Fixations

# Sigma of Gaussian in pixels.
# 33 pixels = ~ 1 visual degree on a 1024x768 monitor at 60 cm.
#
sigma_x = sigma_y = 33.0

# Create x and y pixel ranges for Gauss Mask.
#
x = np.arange(-sigma_x*2.5,sigma_x*2.5, 1)
y = np.arange(-sigma_y*2.5, sigma_y*2.5, 1)

# Create X and Y pixel position values for each element of Gauss. Mask.
#
X, Y = np.meshgrid(x, y)

# Create 2D Gauss Mask as numpy array using X and Y mesh grid data
# and sigma's, with Gauss centered in 2D array (0,0)
#
gauss=mlab.bivariate_normal(X, Y, sigma_x, sigma_y, 0,0)
gauss*=1.0/gauss.max()
ghw,ghh=gauss.shape[1]/2,gauss.shape[0]/2

# Lets plot the Gauss Mask for illustrative purposes...
#
plt.figure()
gmp=plt.imshow(gauss,cmap=cm.gray,origin='lower',extent=(-ghw, ghw,-ghh, ghh))
```

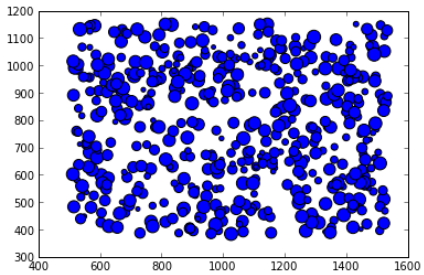


Load Background Image Displayed During Eye Data Collection

In [81]:
```
# Load example background image
# Flip vertically...
image_array=numpy.flipud(mpimg.imread("./images/nature_15.png"))

# Get background image size
image_size=(image_array.shape[1],image_array.shape[0])
ihw,ihh=image_size[0]/2,image_size[1]/2

# Display image for illustrative purposes, 0,0 is image center.
#
plt.figure()
bip=plt.imshow(image_array,origin='lower',extent=(-ihw, ihw,-ihh, ihh))
```

Create some Random Fixation Data

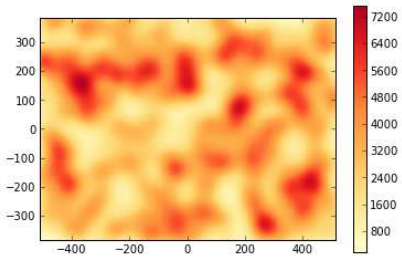
In [82]:
```
# Here, the fixation event data is being simulated as 500 fixations
# of random position within center 50% of fixation_map,
# and with random fixation durations between 150 and 1500 msec.
#
sim_fix_count=500
fix_duration_range=150,1500
fixation_x_range=image_size[0]/2, image_size[0]+image_size[0]/2
fixation_y_range=image_size[1]/2, image_size[1]+image_size[1]/2

# Create the dummy Fixation Data as a 3x500 numpy array
#
fix_pos=np.column_stack( (np.random.randint(*fixation_x_range, size=sim_fix_count),np.random.randint(*fixation_y_range, size=sim_fix_count),np

# Plot the fixation data for illustrative purposes.
#
x = [0,2,4,6,8,10];
y = [0]*len(x);
s = [20*2**n for n in range(len(x))];
plt.figure()
sfd=plt.scatter(fix_pos[:,0],fix_pos[:,1],s=fix_pos[:,2]/10)
```



Create the Fixation Density Map Layer based on the Gauss Mask Template and the Fixation Data

In [83]:
```
# Start with empty 2D numpy array 2x size of fixation data
# (makes applying Gauss mask for each fixation easier)
#
fixation_map=np.zeros((image_size[1]*2, image_size[0]*2))

# Apply Gauss. Mask to each fixation position based on the created fixation event data.
#
for fx,fy,fix_dur in fix_pos:
    fixation_map[fy-ghh:fy+ghh+1,fx-ghw:fx+ghh+1]+=(gauss*fix_dur)

# Trim the Fixation Density Mask Back Down to Match the Image Size
#
fixation_map=fixation_map[image_size[1]/2:image_size[1]+image_size[1]/2,image_size[0]/2:image_size[0]+image_size[0]/2]


# Plot Fixation Density Mask Alone for illustrative purposes
# using a Yellow->Orange->Red Color Map.
clmap=brewer2mpl.get_map('YlOrRd', 'sequential', 8).mpl_colormap
im = plt.imshow(fixation_map, interpolation='nearest', origin='lower',extent=(-image_size[0]/2, image_size[0]/2,-image_size[1]/2, image_size[1

plt.colorbar()
```

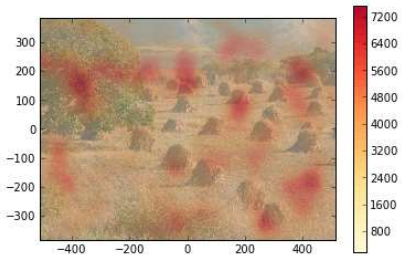
Out[83]: <matplotlib.colorbar.Colorbar instance at 0x63639710>



Putting it all Together: Heat Map Representation of Fixation Position and Dwell Time Density During Image Viewing

In [84]:
```
plt.figure()

# Draw the background Image
#
plt.imshow(image_array,interpolation='nearest',origin='lower',extent=(-image_size[0]/2, image_size[0]/2,-image_size[1]/2, image_size[1]/2))


# Create RGBA values for the color map created above.
#
clmap._init()

# Set the Color Map Transparency to Increase as a Function of Fixation Dwell Time.
#
clmap._lut[:,-1]=np.linspace(0.5, .8, clmap.N+3)

# Draw the Fixation Map Mask over the Background Image using the Color Map:
#
im = plt.imshow(fixation_map, interpolation='nearest', origin='lower',
                extent=(-image_size[0]/2, image_size[0]/2,-image_size[1]/2, image_size[1]/2),cmap=clmap)#,alpha=0.5)

# Display the Heat Map Scale:
#
plt.colorbar()
```

Out[84]: <matplotlib.colorbar.Colorbar instance at 0x62F357D8>



That's it!