

Professor_Bear_Image_Analysis_Morphological_Operations

March 14, 2017

1 Professor Bear :: Image Analysis :: Morphological Operations

1.1 Professor Bear github

Code for Professor Bear YouTube videos at <https://github.com/nikbearbrown>

1.2 Download Anaconda 4 for Python 2.7

Download Anaconda 4 for Python 2.7 version <https://www.continuum.io/downloads>

Anaconda 4.3.0 includes an easy installation of Python (2.7.13, 3.4.5, 3.5.2, and/or 3.6.0) and updates of over 100 pre-built and tested scientific and analytic Python packages. These packages include NumPy, Pandas, SciPy, Matplotlib, and Jupyter. Over 620 more packages are available.
<https://docs.continuum.io/anaconda/pkg-docs>

1.3 iPython

Go to the directory that has your iPython notebook

At the command line type

jupyter notebook notebookname

ipython notebook notebookname will also work

For example,

jupyter notebook Professor_Bear_Image_Analysis_Loading_Histograms.ipynb

```
In [25]: # Bring in python image analysis libraries
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
from skimage import color
import skimage.filters as filters
from skimage.transform import hough_circle
from skimage.feature import peak_local_max
from skimage import feature
from skimage import morphology
from skimage.draw import circle_perimeter
from skimage import img_as_float, img_as_ubyte
from skimage import segmentation as seg
from skimage.morphology import watershed
```

```

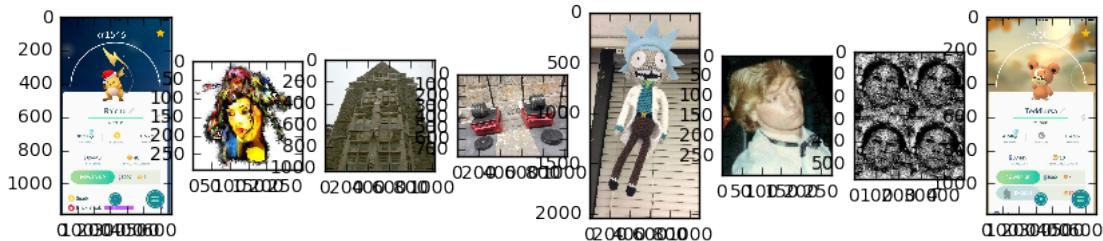
from scipy import ndimage as nd
from scipy.ndimage import convolve
from skimage import feature
import glob # for bulk file import

# Set defaults
plt.rcParams['image.cmap'] = 'gray' # Display grayscale images in... grays
plt.rcParams['image.interpolation'] = 'none' # Use nearest-neighbour
plt.rcParams['figure.figsize'] = 10, 10

# Import test images
imgpaths = glob.glob("./img/*.jpg") + glob.glob("./img/*.png")
# imgpaths = glob.glob("images/*.jpg") + glob.glob("images/*.png") Windows
# Windows has different relative paths than Mac/Unix
imgset = [mpimg.imread(x) for x in imgpaths]

# Display thumbnails of the images to ensure loading
plt.figure()
for i,img in enumerate(imgset):
    plt.subplot(1, len(imgset), i+1)
    plt.imshow(img, cmap = 'gray')

```



1.4 Morphological Operations

Morphological operations (https://en.wikipedia.org/wiki/Mathematical_morphology) are a set of image processing operations that change images based on shapes. The number of pixels added or removed from the objects in an image depends on the size and shape used to process the image. The basic idea in morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple “probe” is called the structuring element.

```

In [26]: # Applies a morphological operator to remove noise
def morphological_denoise(img, shape, size):
    if shape == 'square':
        kernel = morphology.square(width=size)
    elif shape == 'diamond':
        kernel = morphology.diamond(radius=size)

```

```

    else:
        print("Shape must be 'square' or 'diamond'.")
        return None
    return morphology.opening(img, kernel)

```

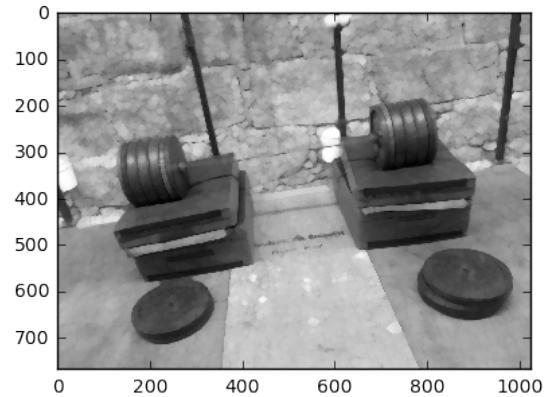
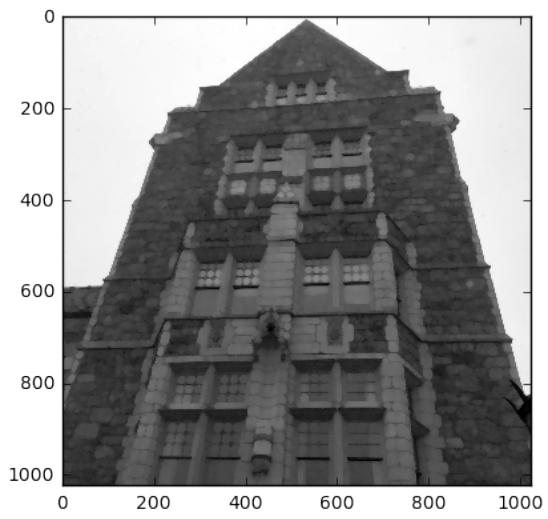
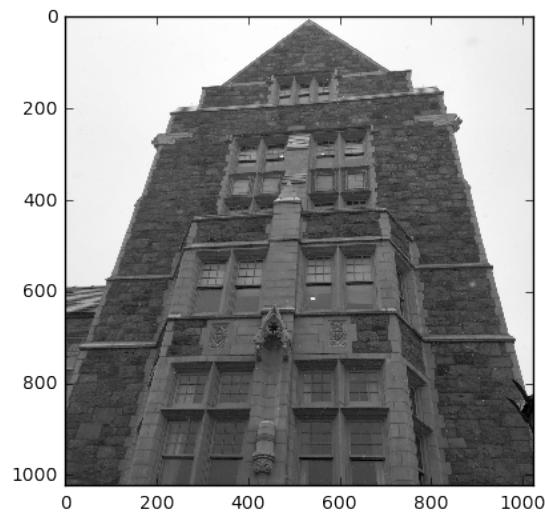
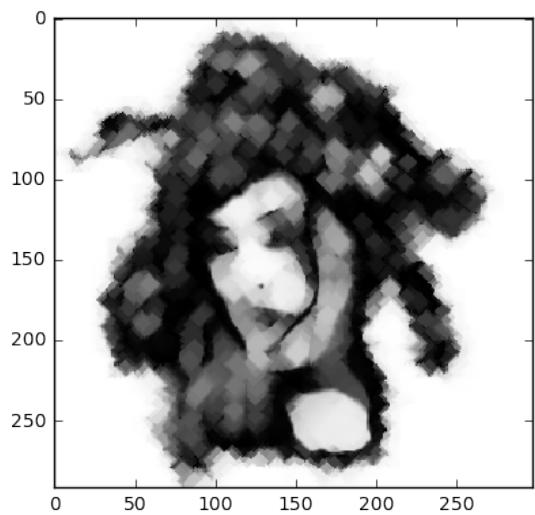
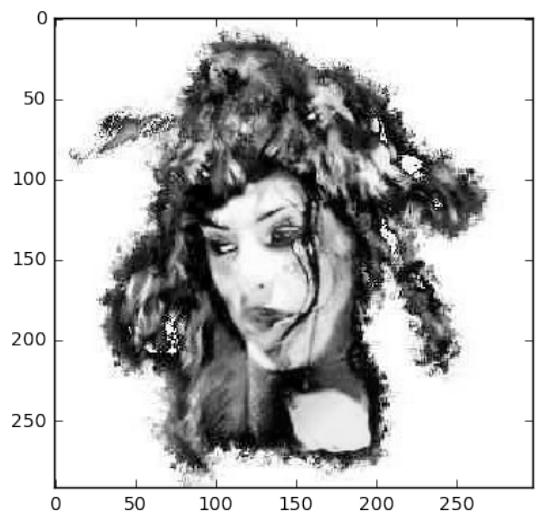
In [27]: # Apply to image set

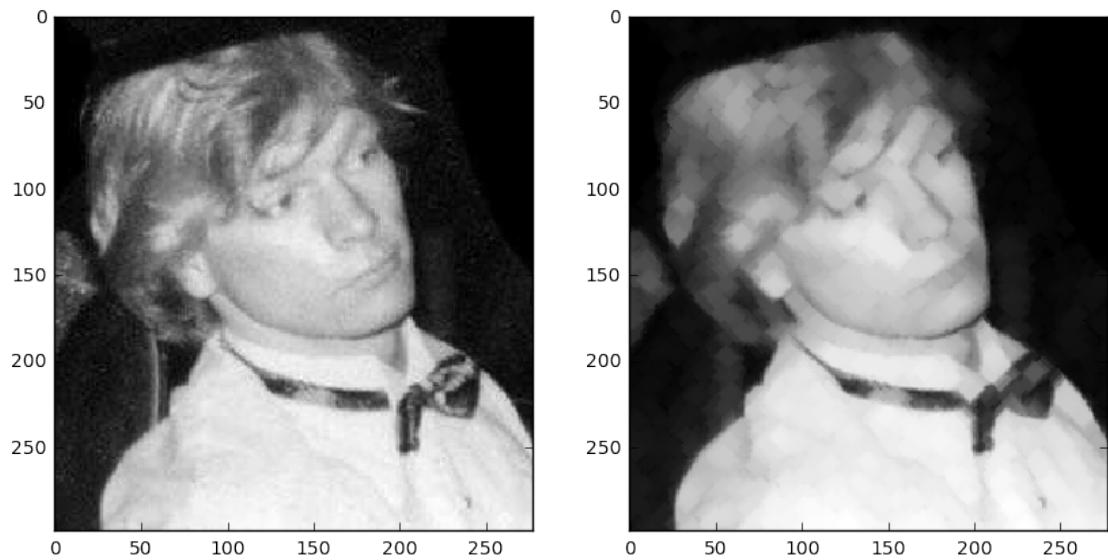
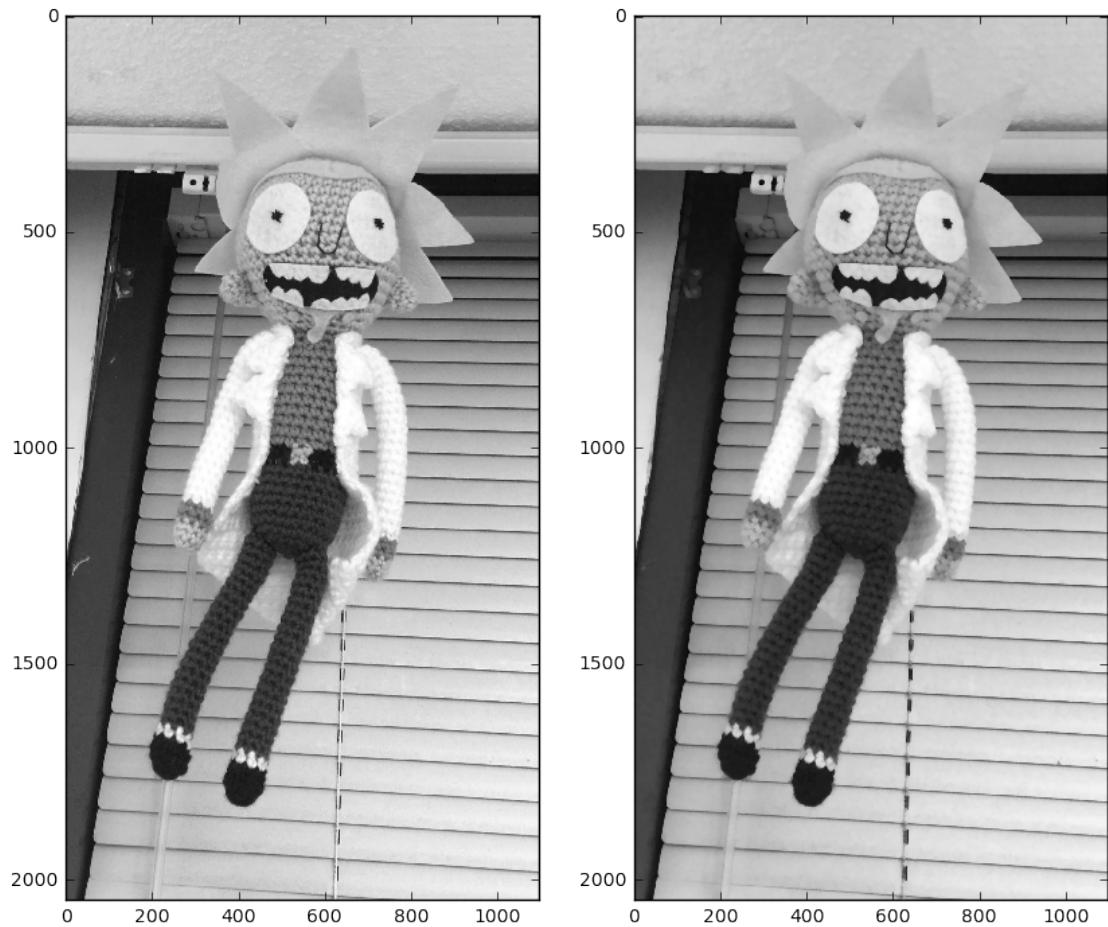
```

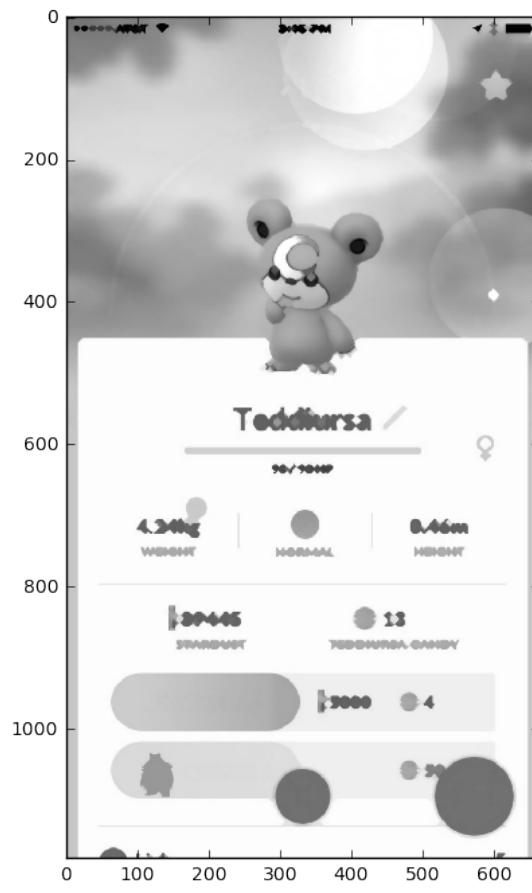
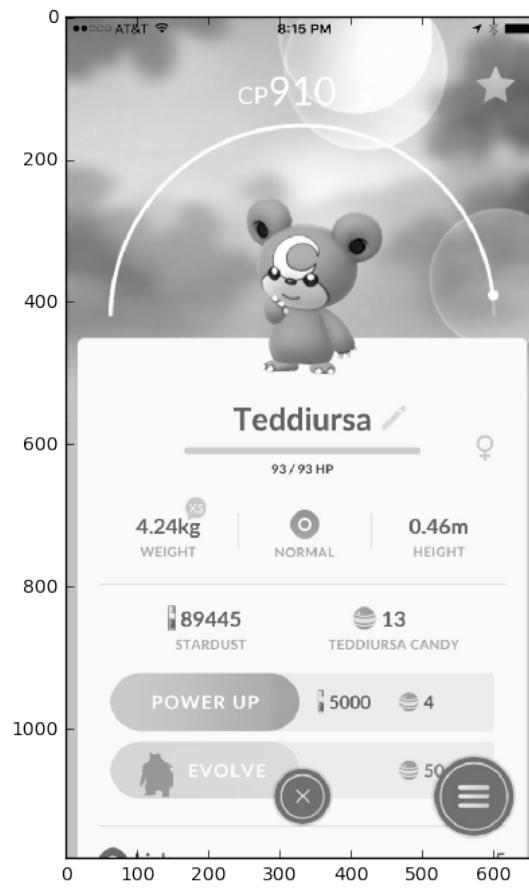
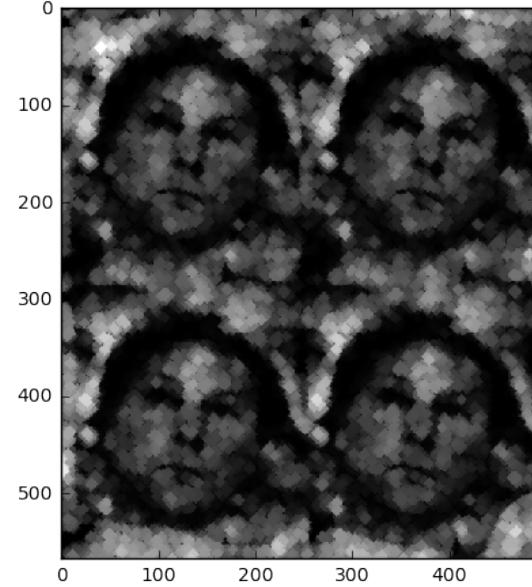
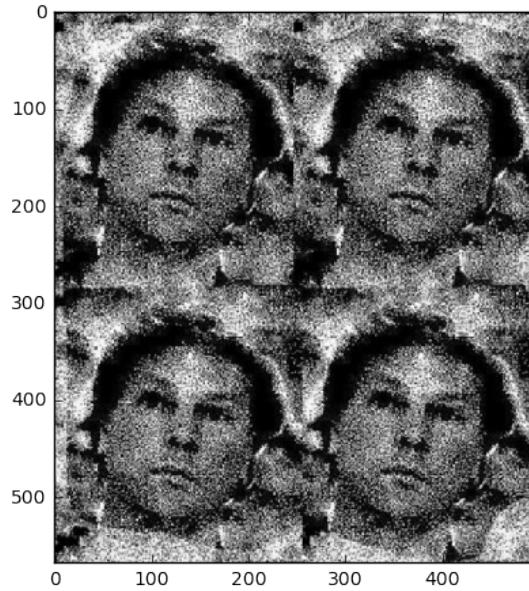
size=5.0
for i,img in enumerate(imgset):
    imgbw = img_as_float(color.rgb2grey(img))
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.imshow(imgbw)
    plt.subplot(1, 2, 2)
    plt.imshow(morphological_denoise(imgbw, 'diamond', size))

```



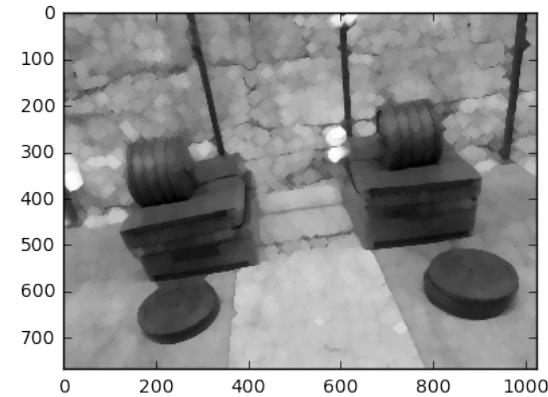
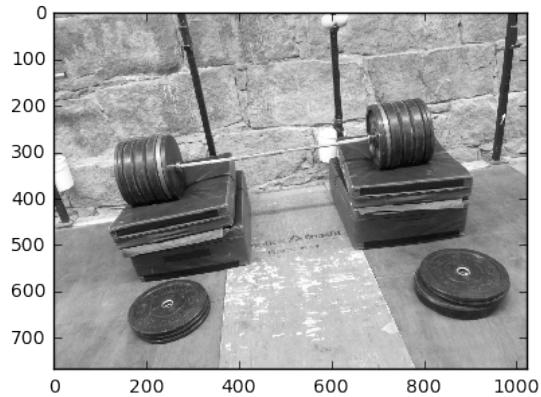
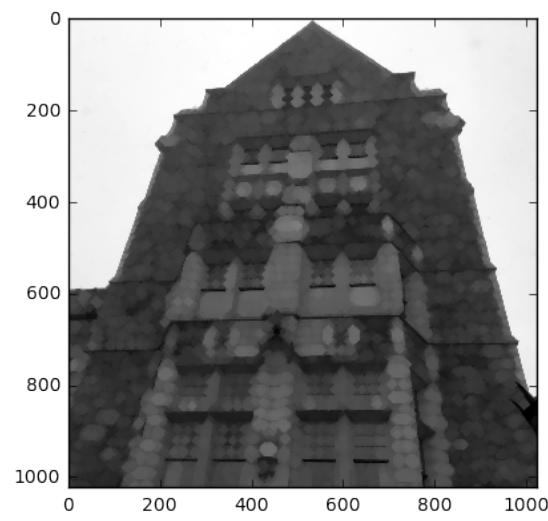
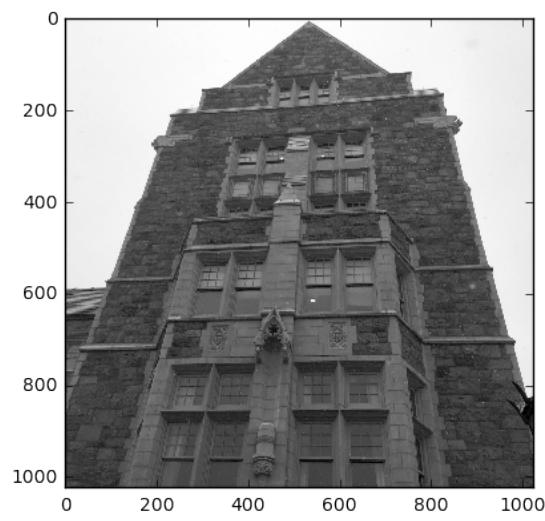
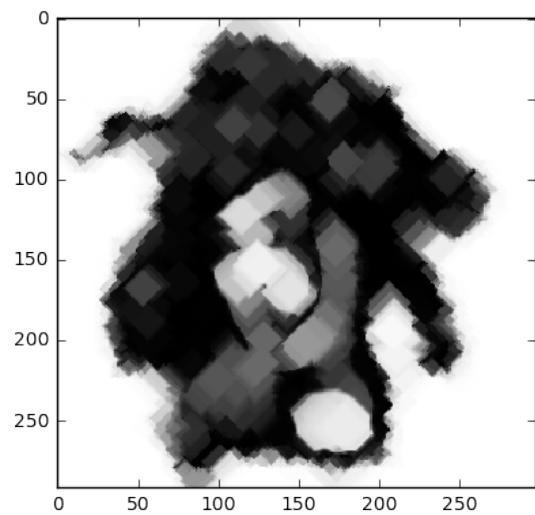
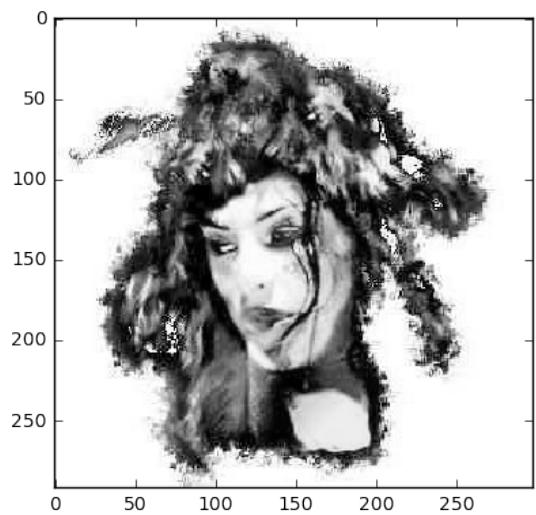


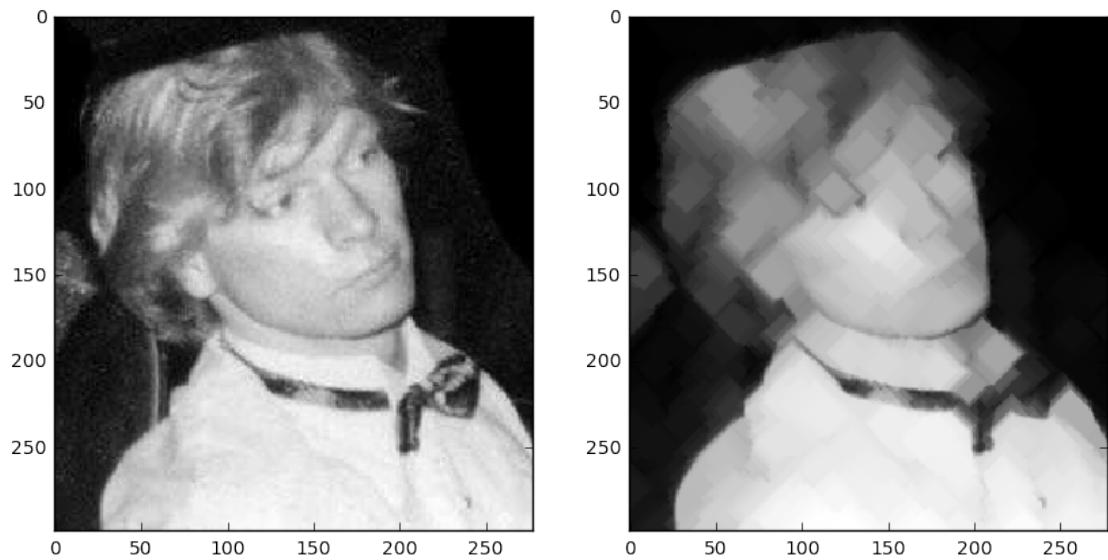
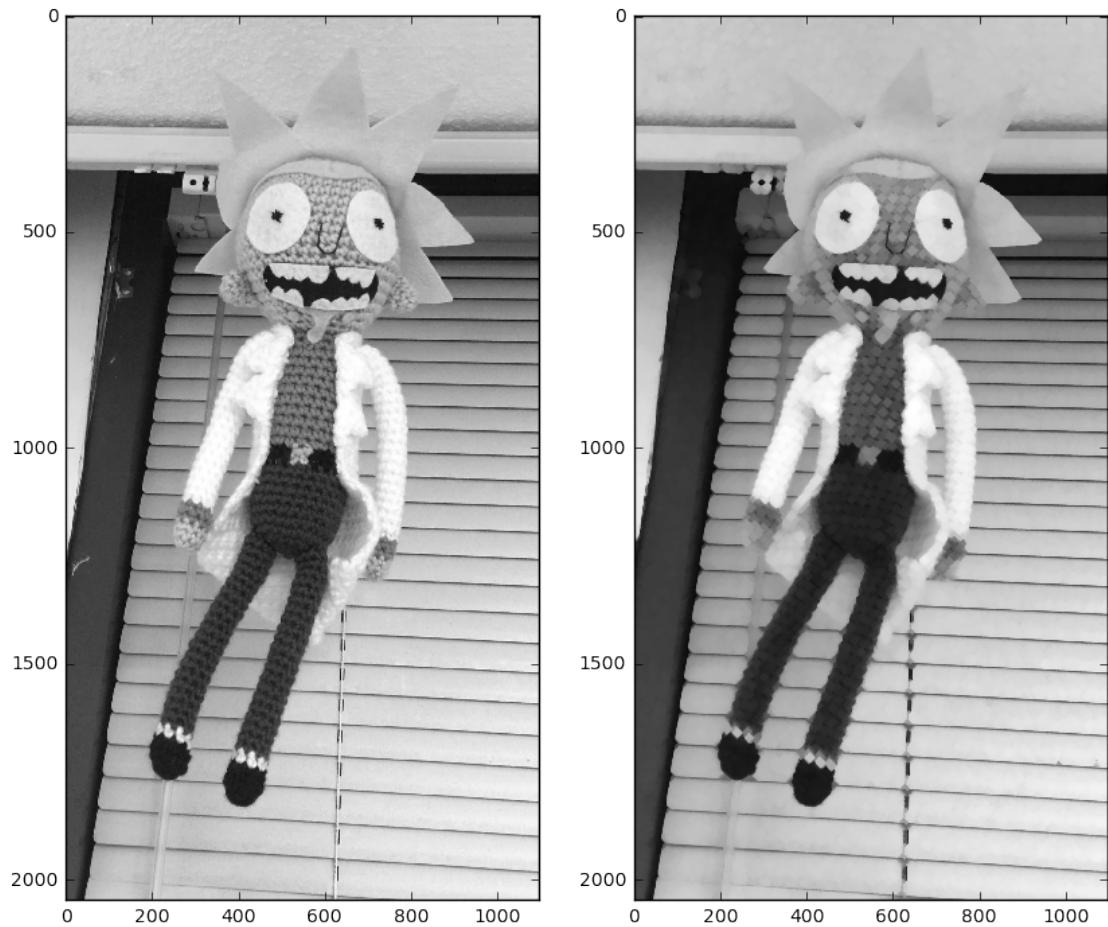


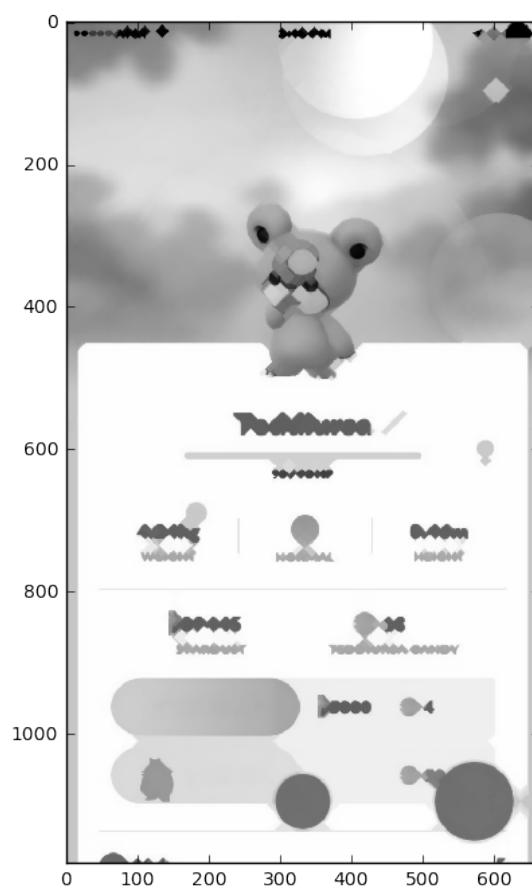
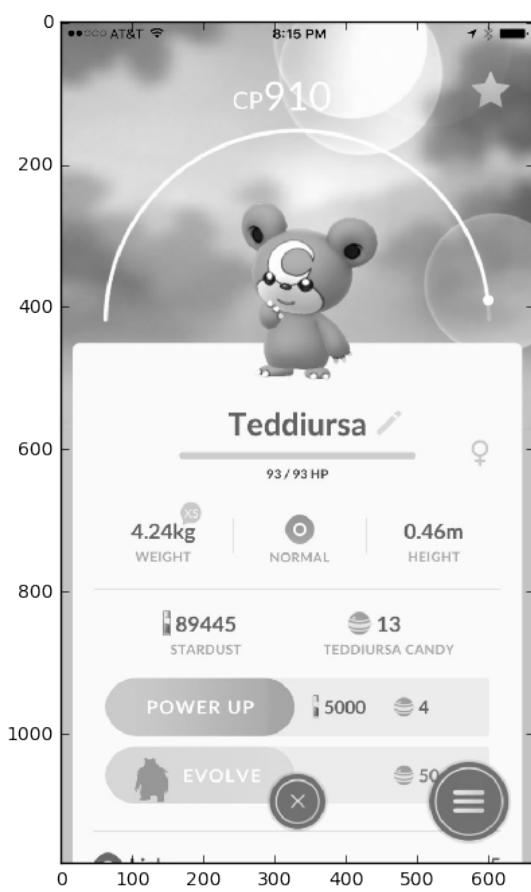
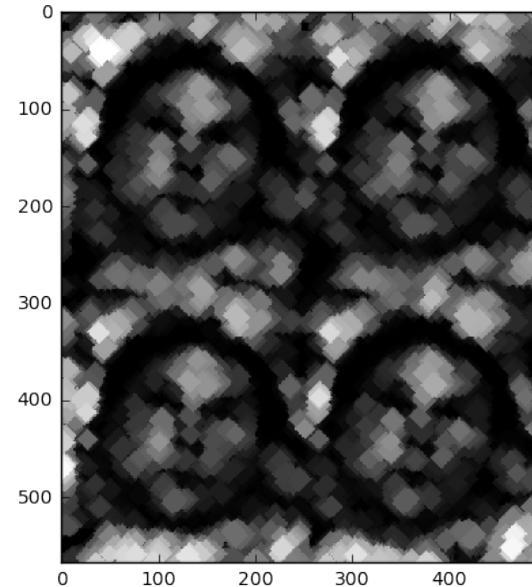
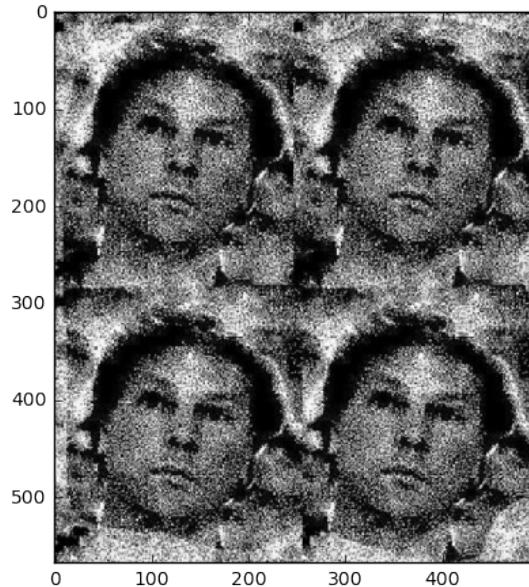


```
In [28]: # Apply to image set
size=10.0
for i,img in enumerate(imgset):
    imgbw = img_as_float(color.rgb2grey(img))
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.imshow(imgbw)
    plt.subplot(1, 2, 2)
    plt.imshow(morphological_denoise(imgbw, 'diamond', size))
```



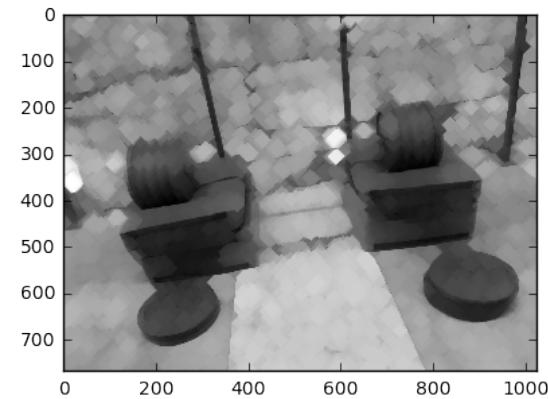
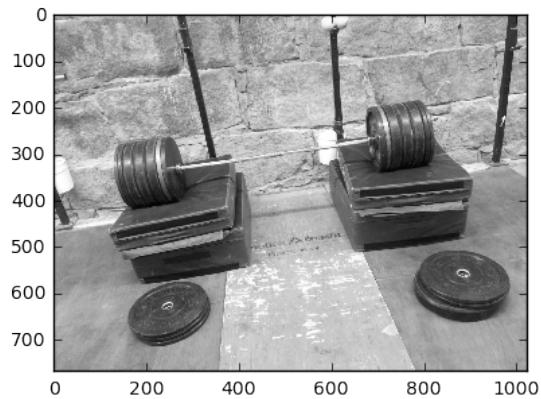
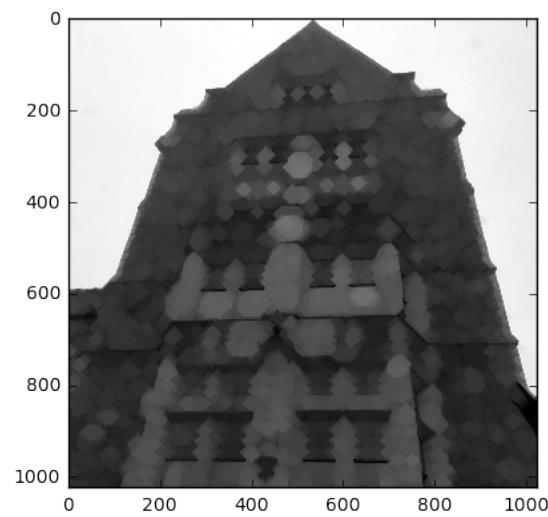
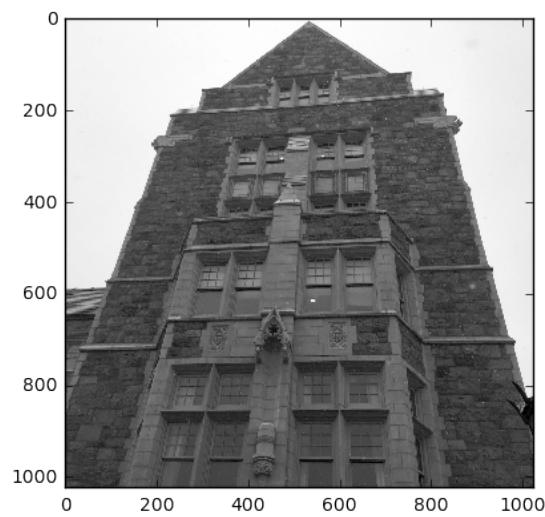
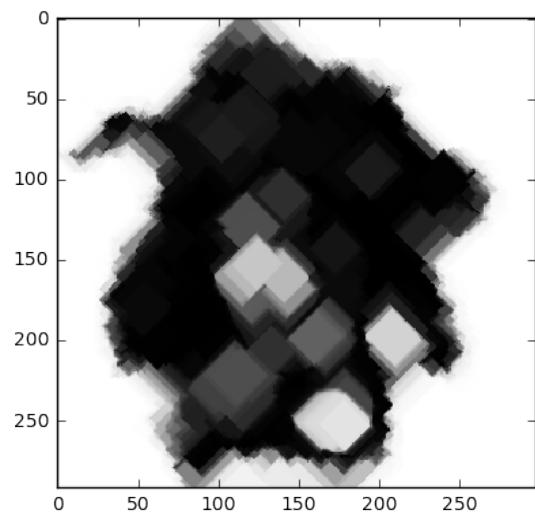
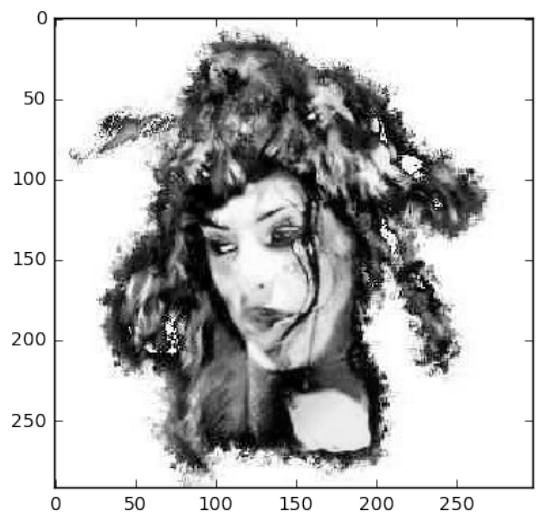


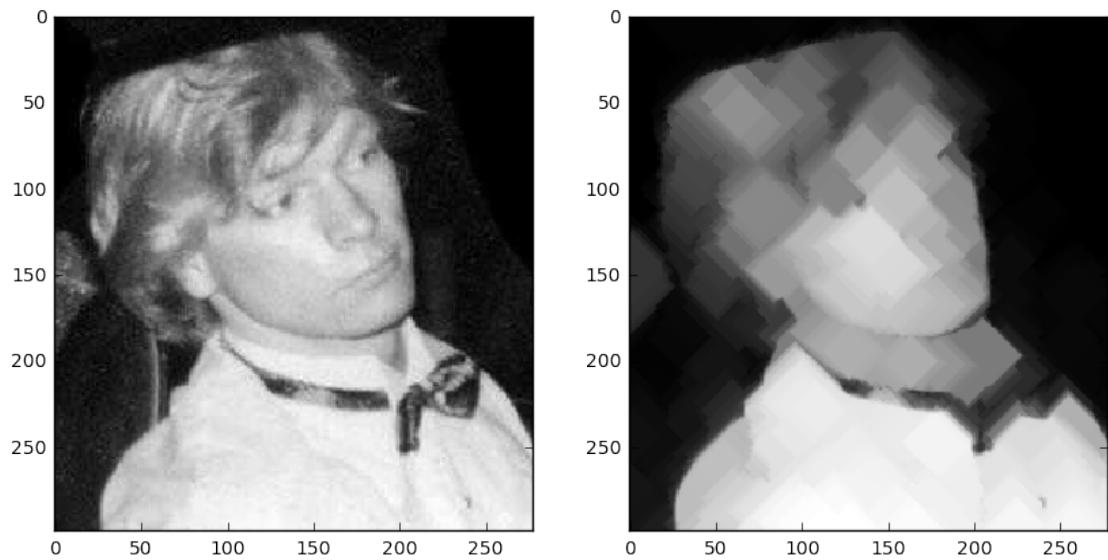
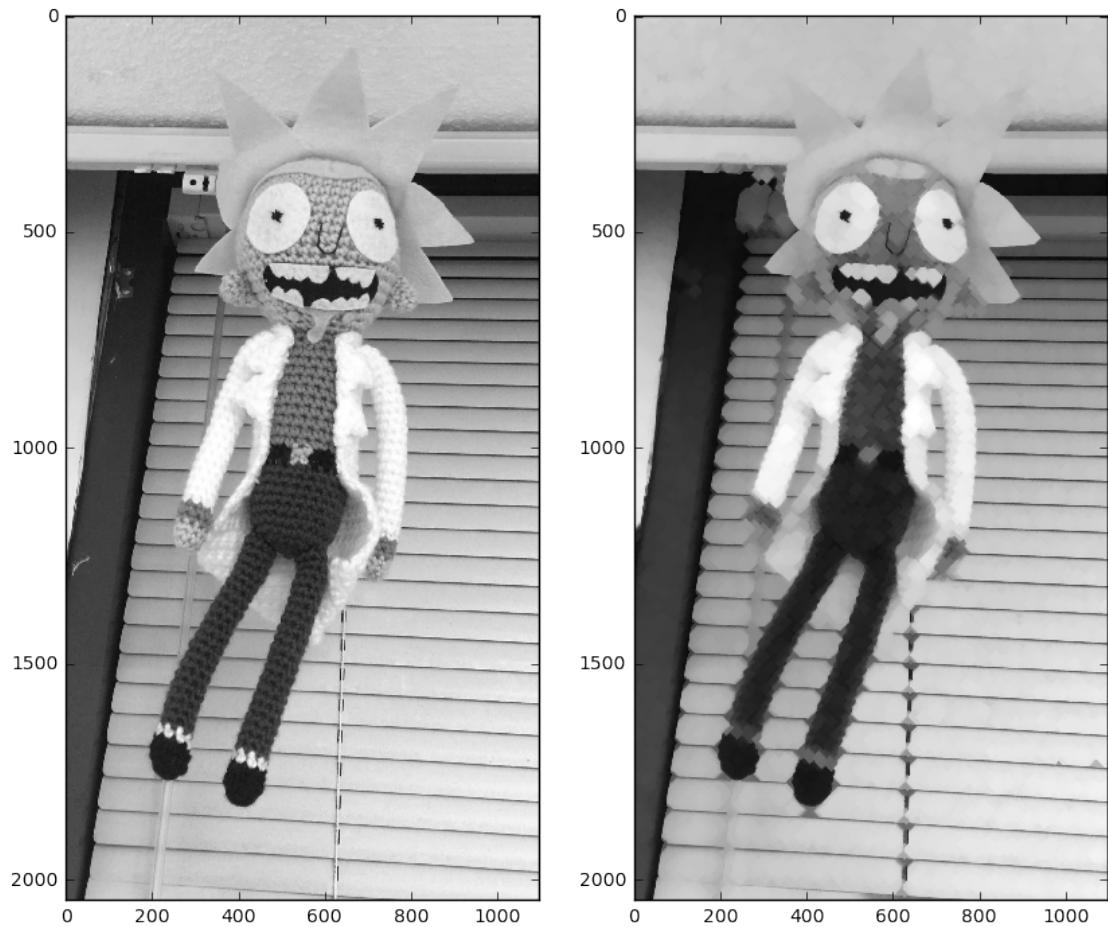


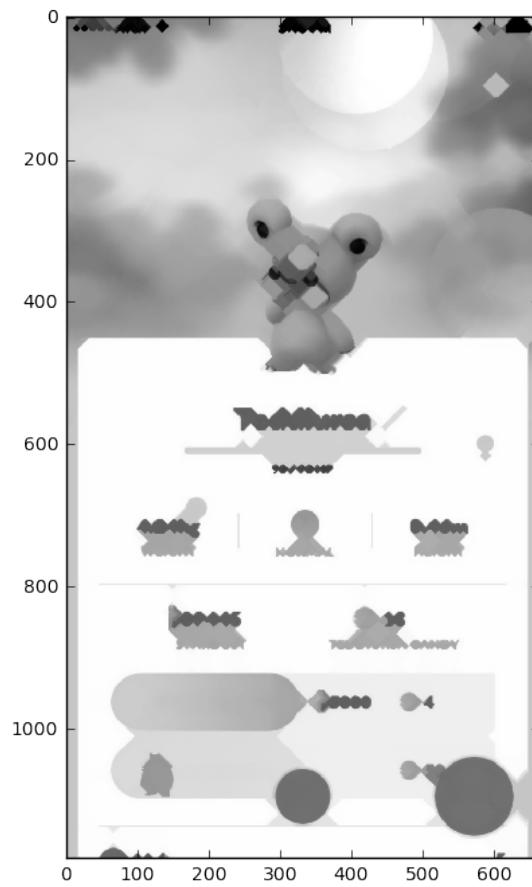
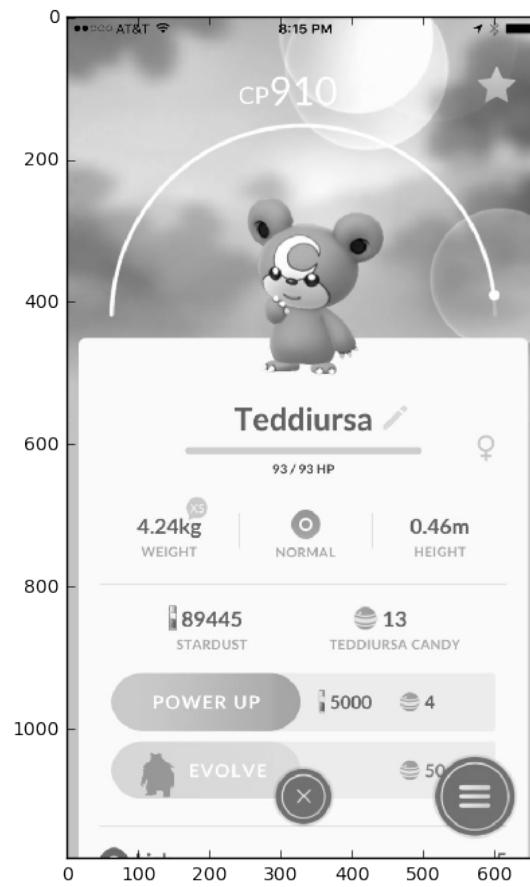
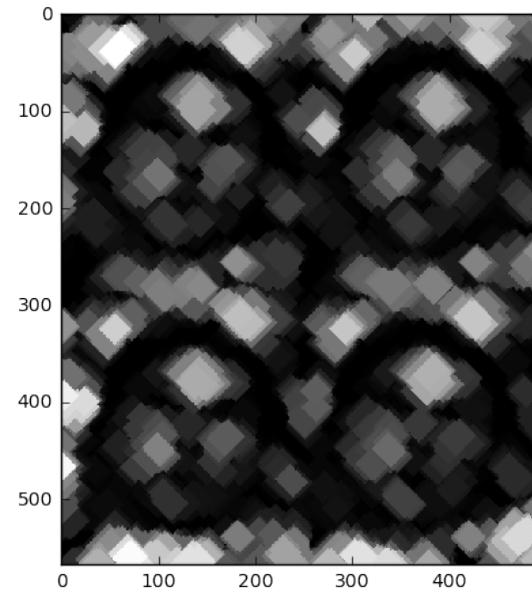
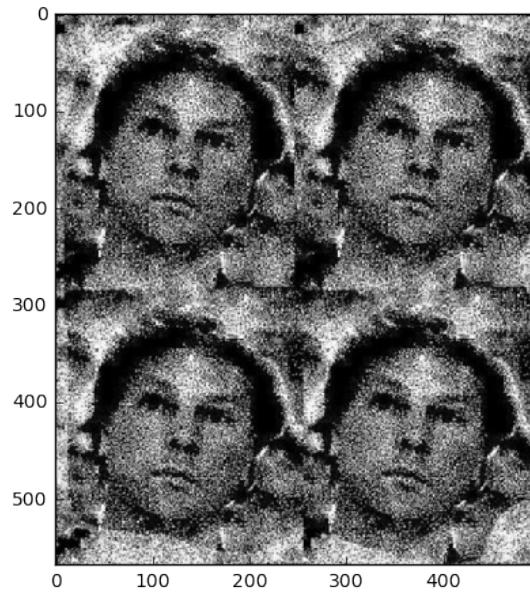


```
In [29]: # Apply to image set
size=15.0
for i,img in enumerate(imgset):
    imgbw = img_as_float(color.rgb2grey(img))
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.imshow(imgbw)
    plt.subplot(1, 2, 2)
    plt.imshow(morphological_denoise(imgbw, 'diamond', size))
```



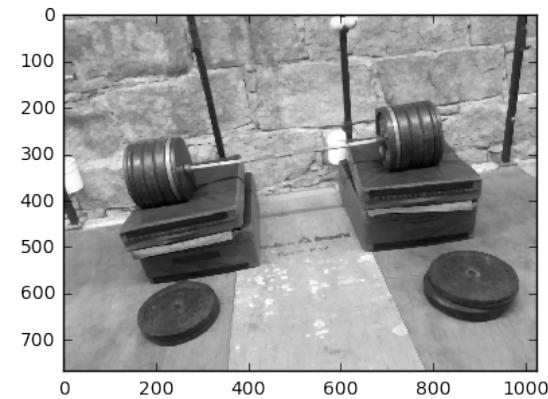
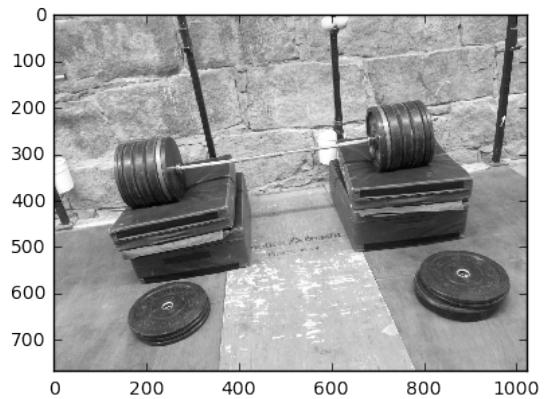
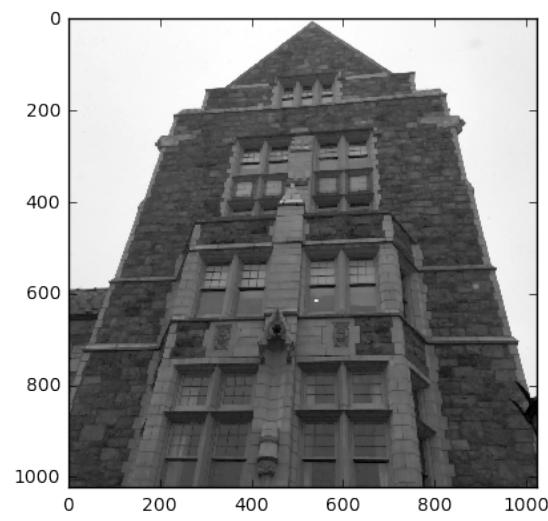
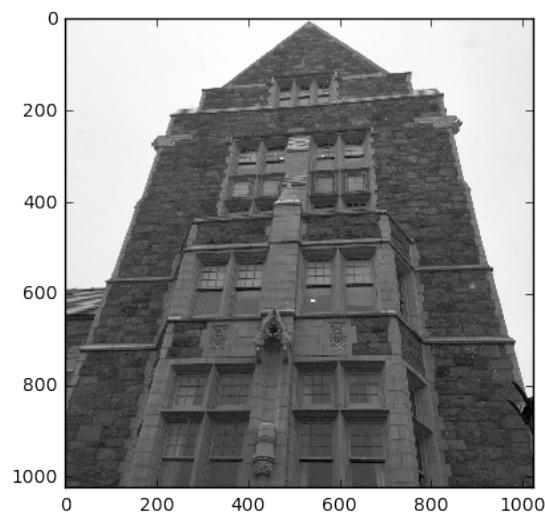
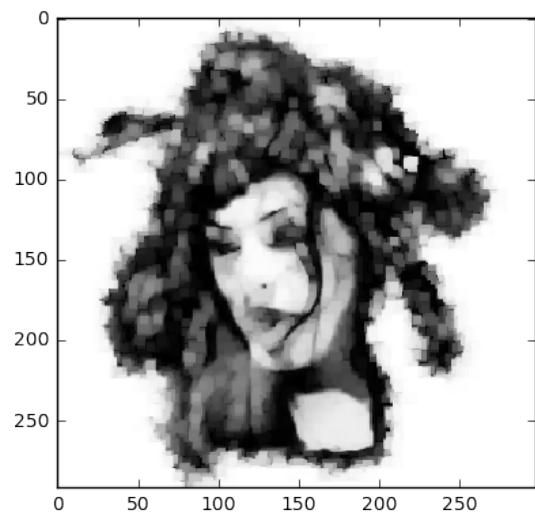
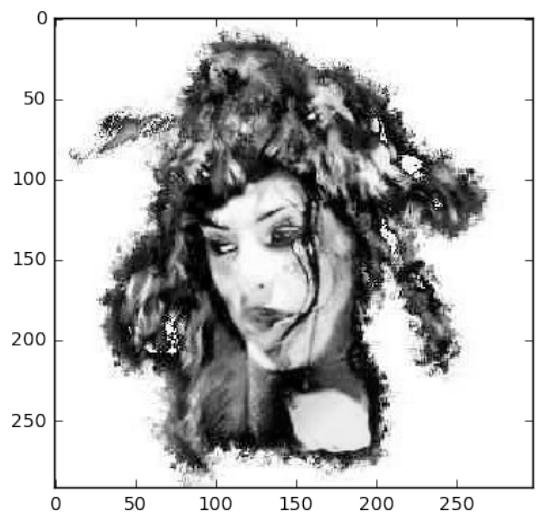


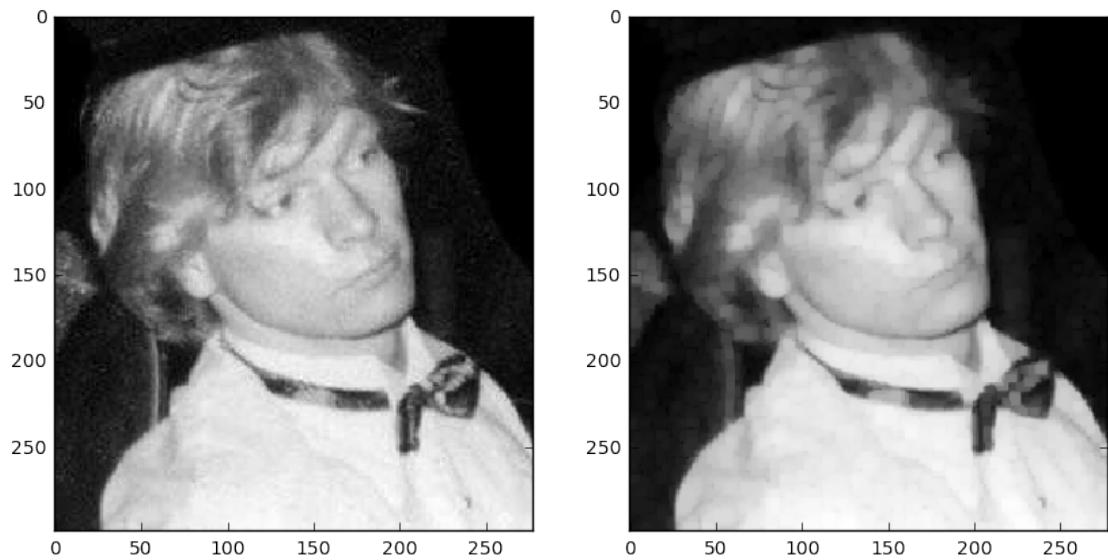
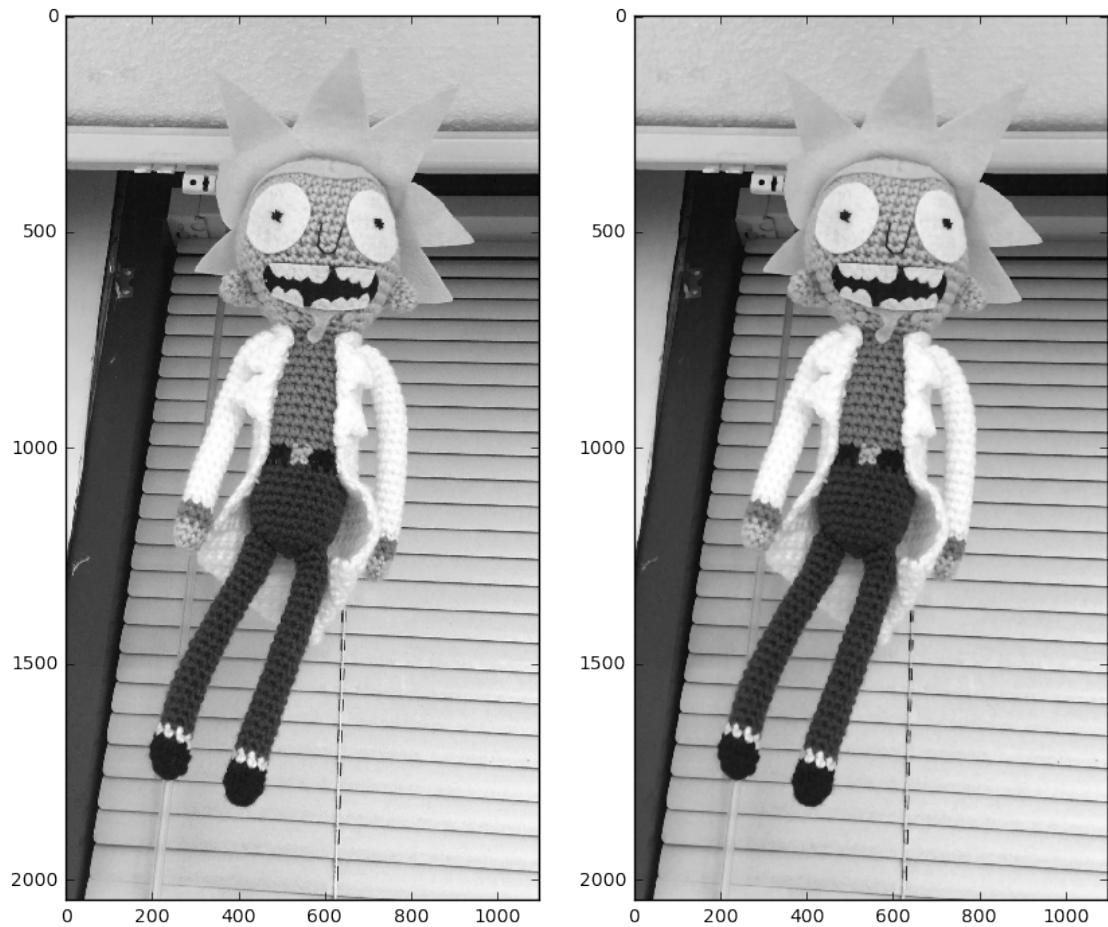


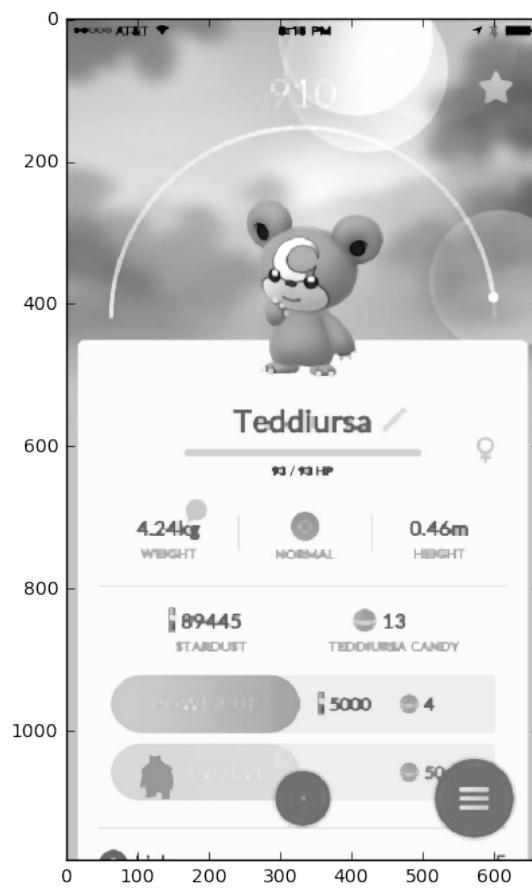
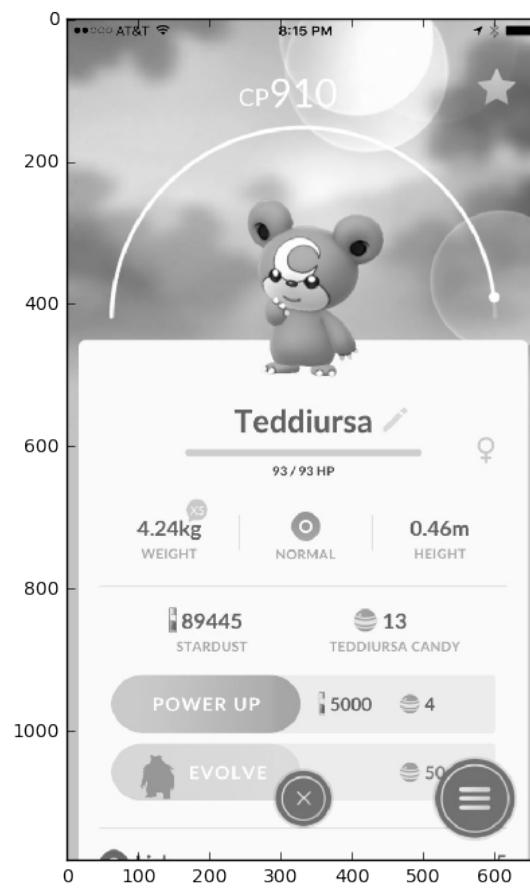
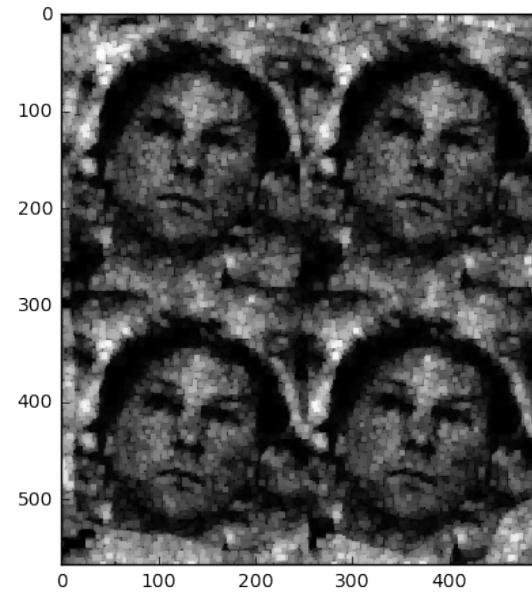
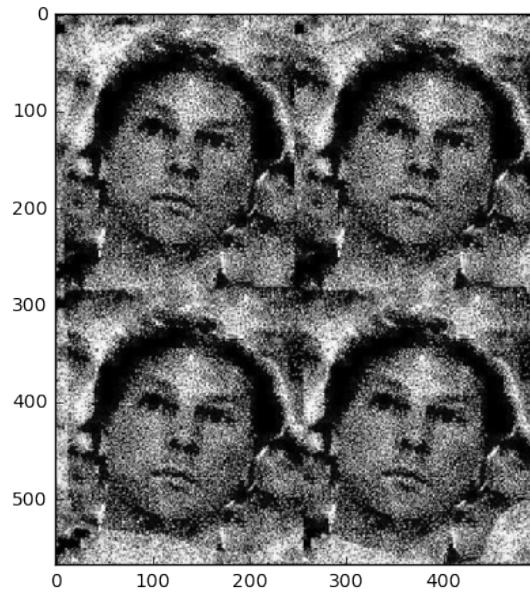


```
In [30]: # Apply to image set
size=5
for i,img in enumerate(imgset):
    imgbw = img_as_float(color.rgb2grey(img))
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.imshow(imgbw)
    plt.subplot(1, 2, 2)
    plt.imshow(morphological_denoise(imgbw, 'square', size))
```



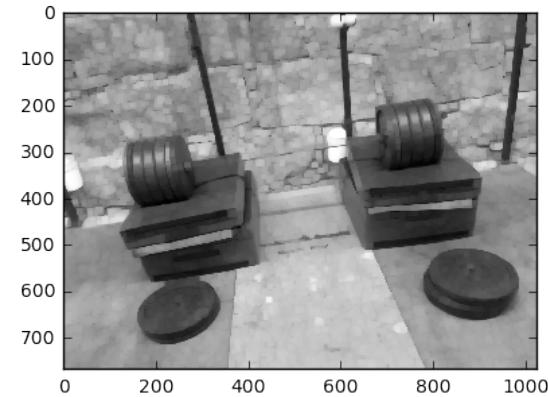
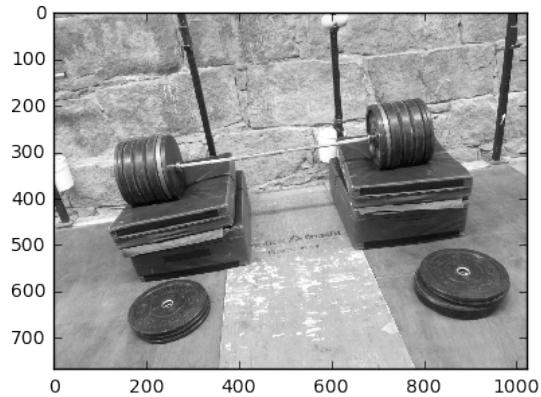
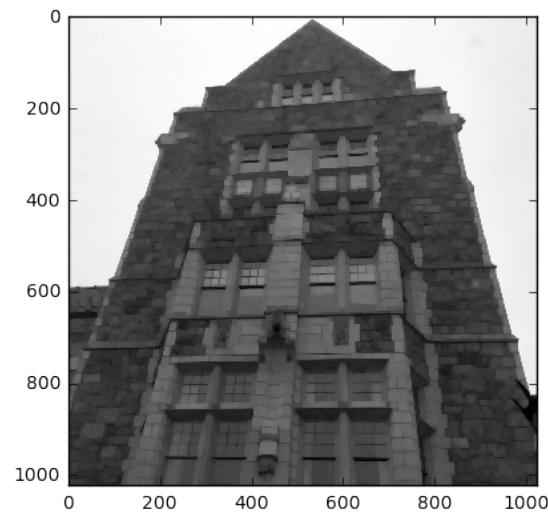
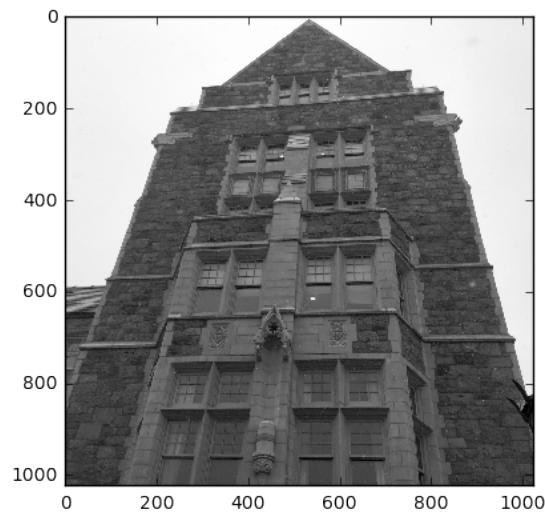
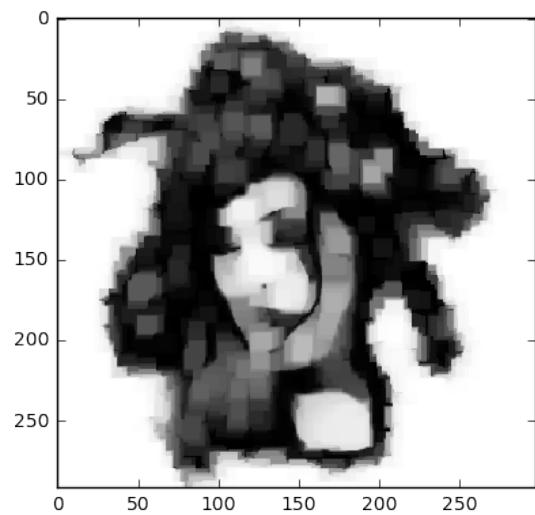
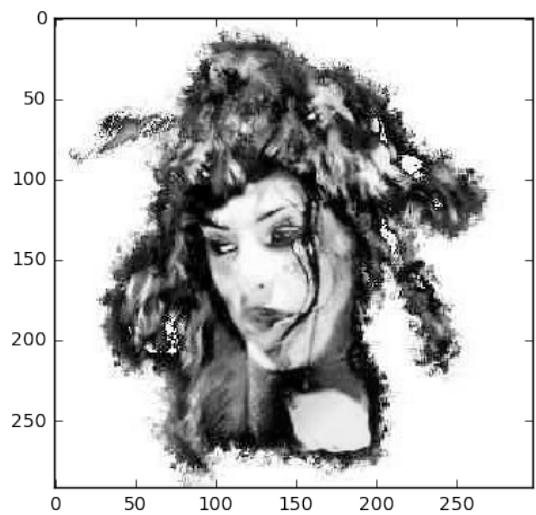


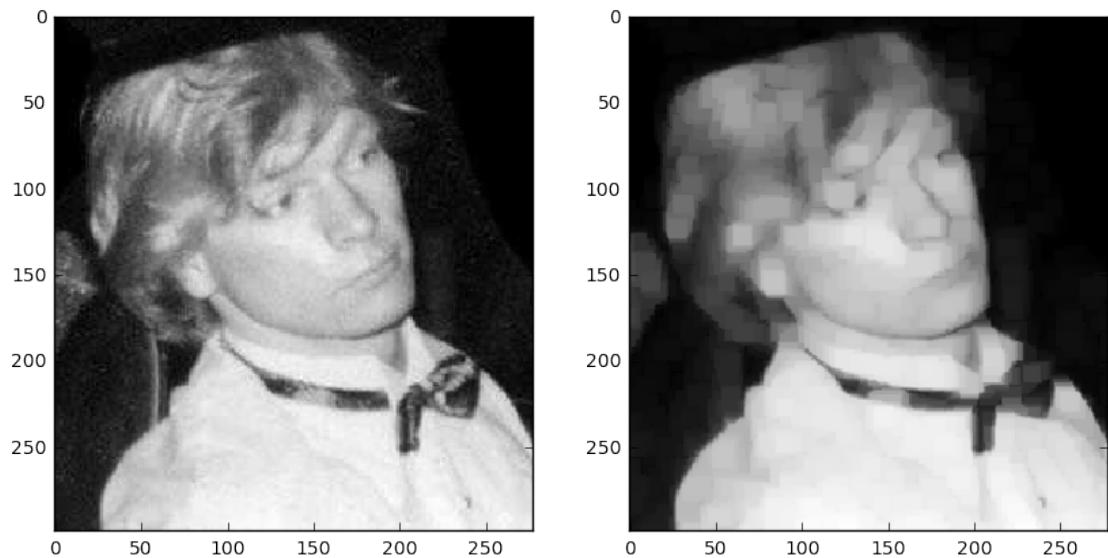
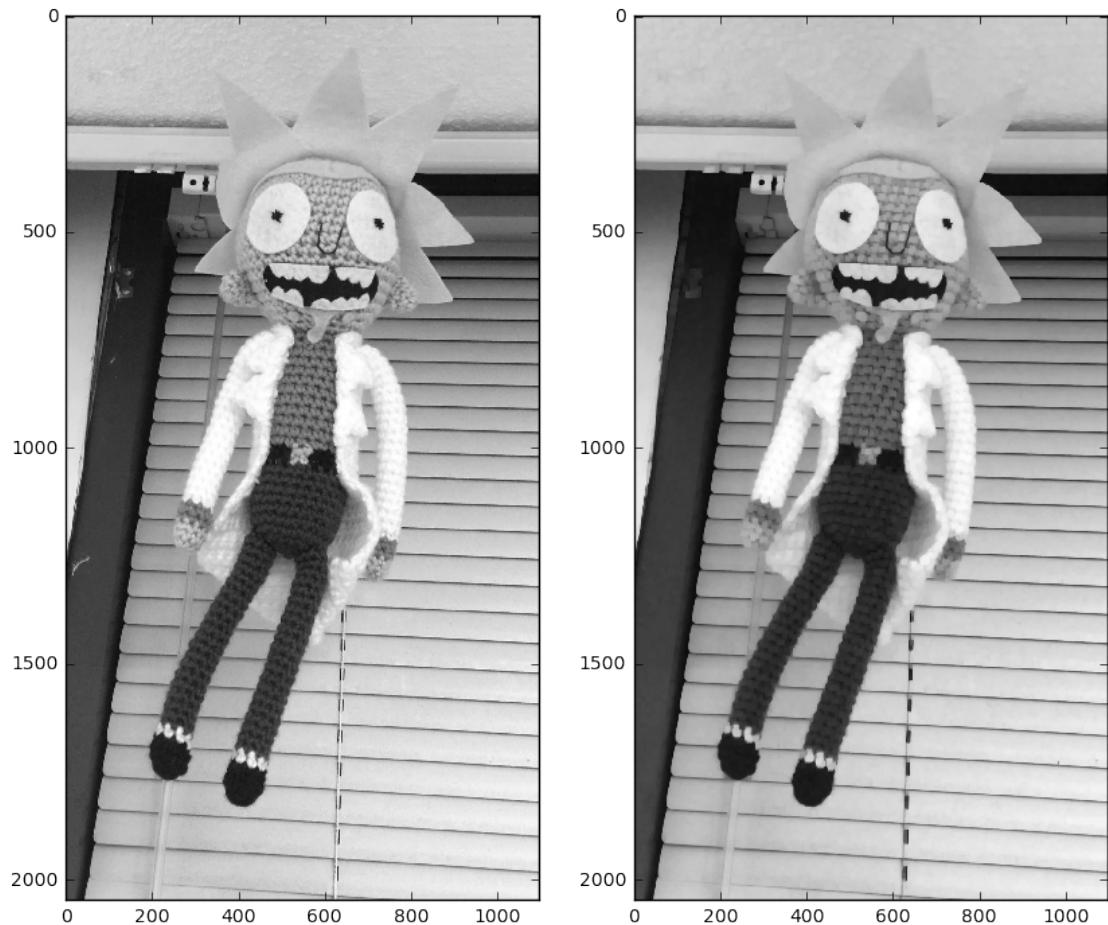


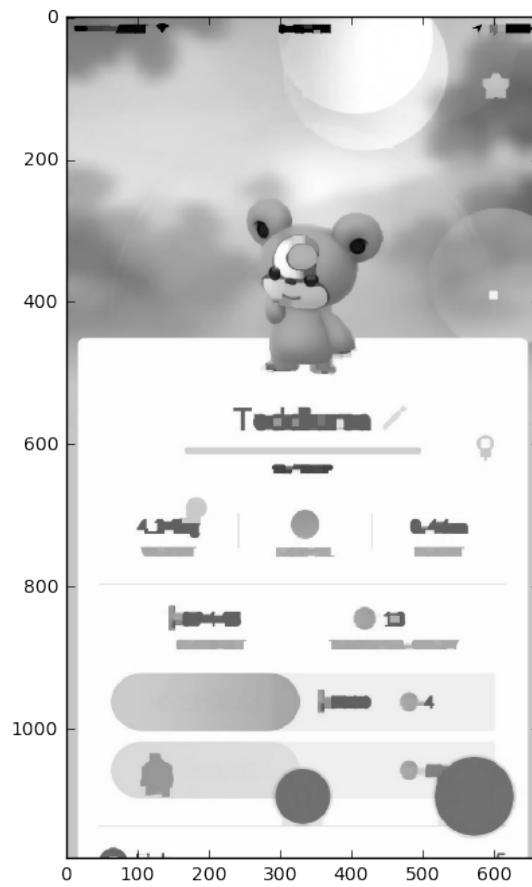
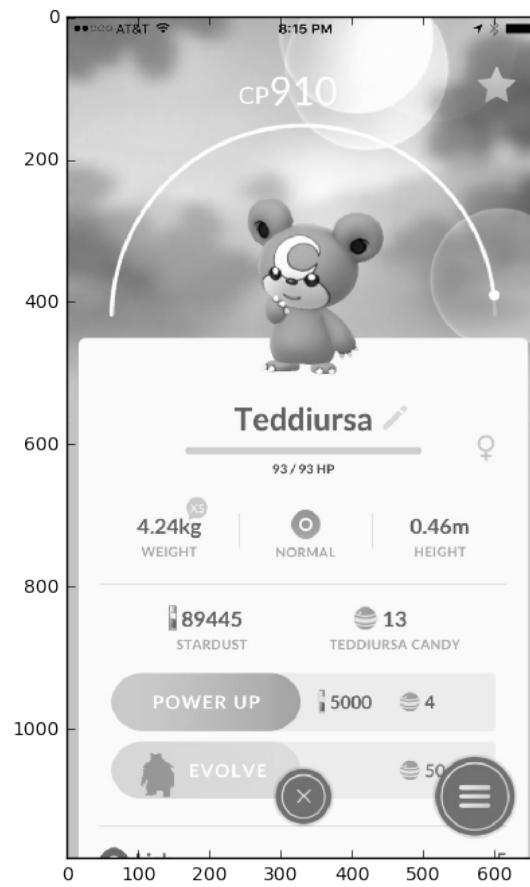
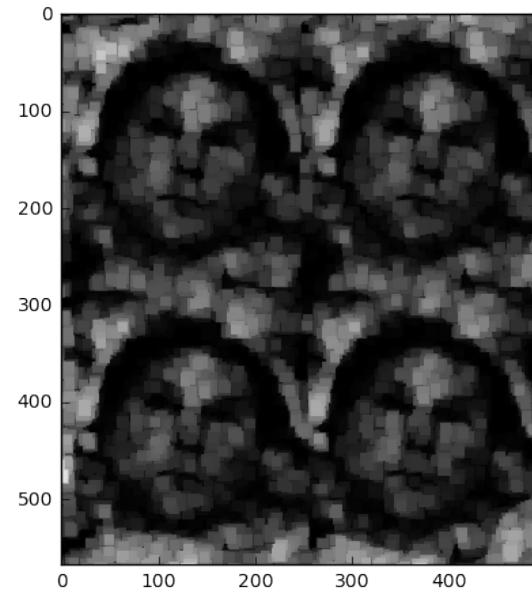
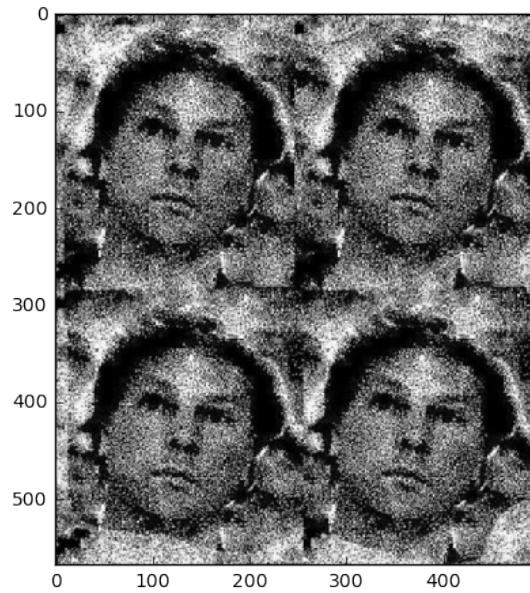


```
In [31]: # Apply to image set
size=10
for i,img in enumerate(imgset):
    imgbw = img_as_float(color.rgb2grey(img))
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.imshow(imgbw)
    plt.subplot(1, 2, 2)
    plt.imshow(morphological_denoise(imgbw, 'square', size))
```



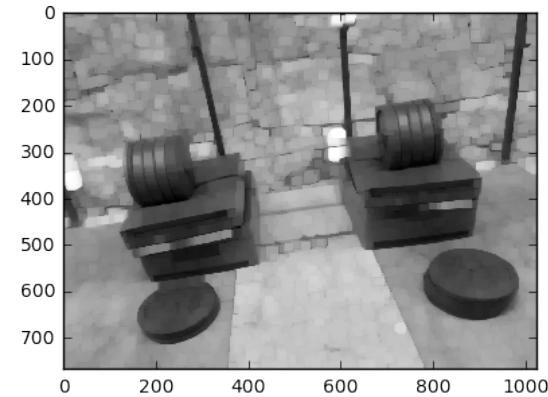
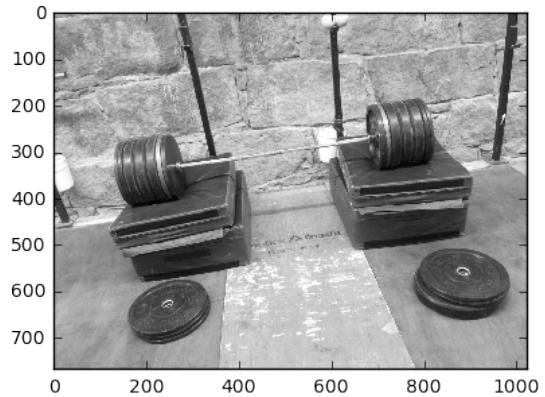
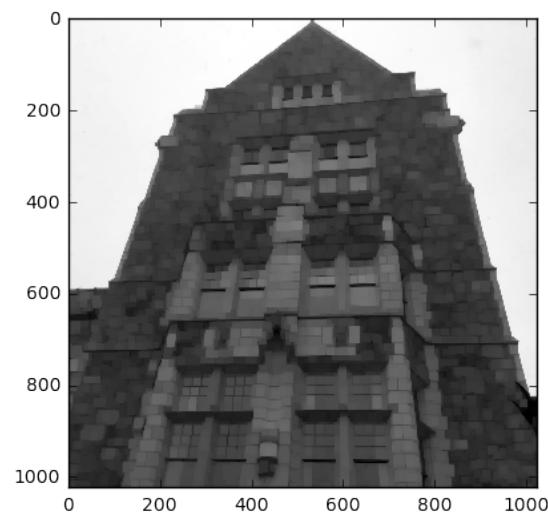
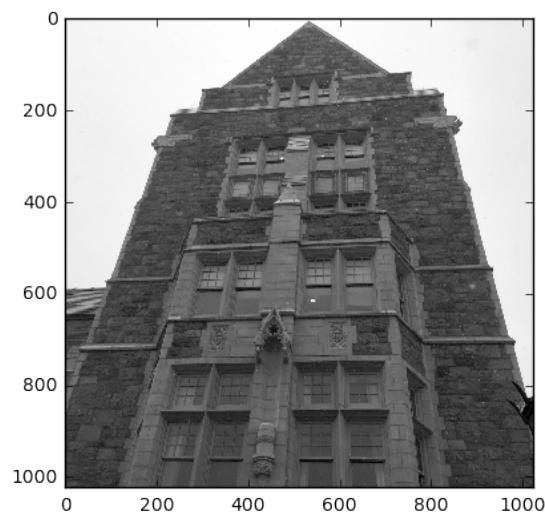
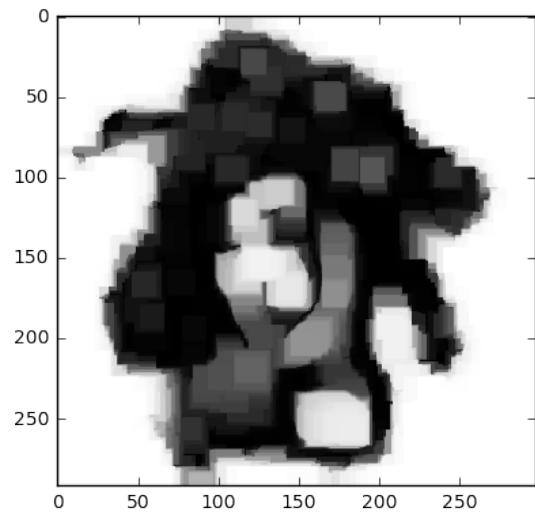
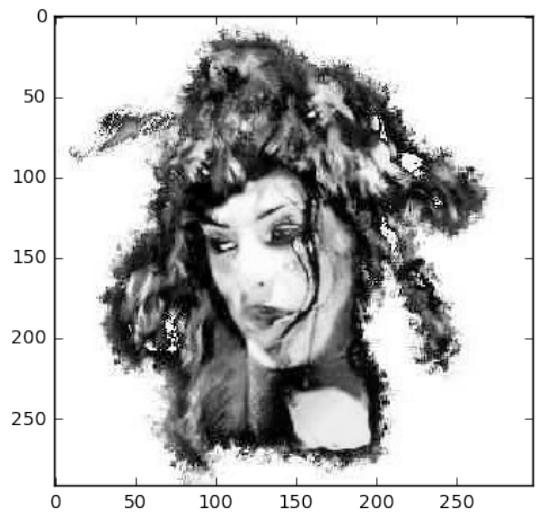


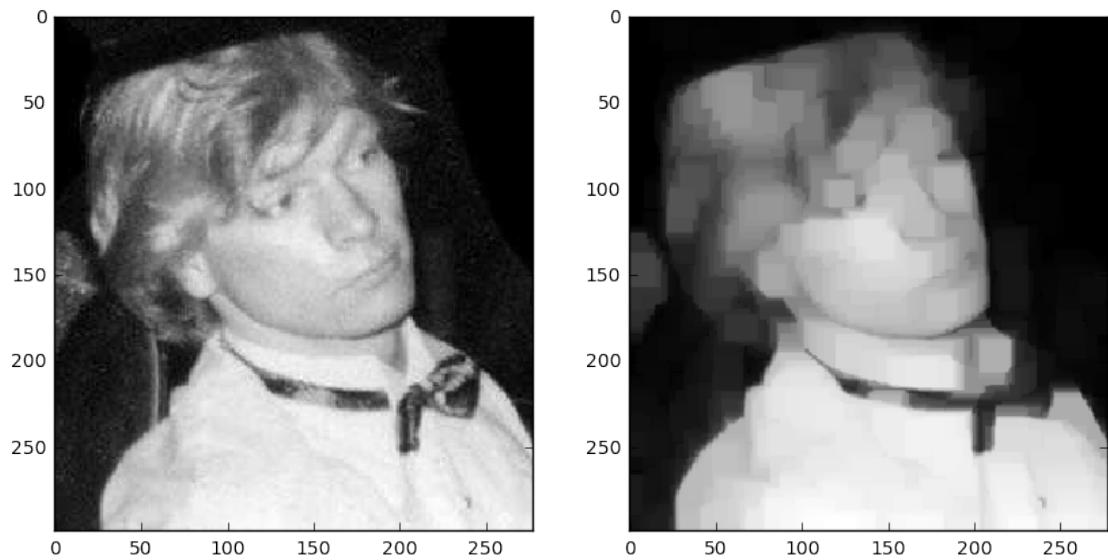
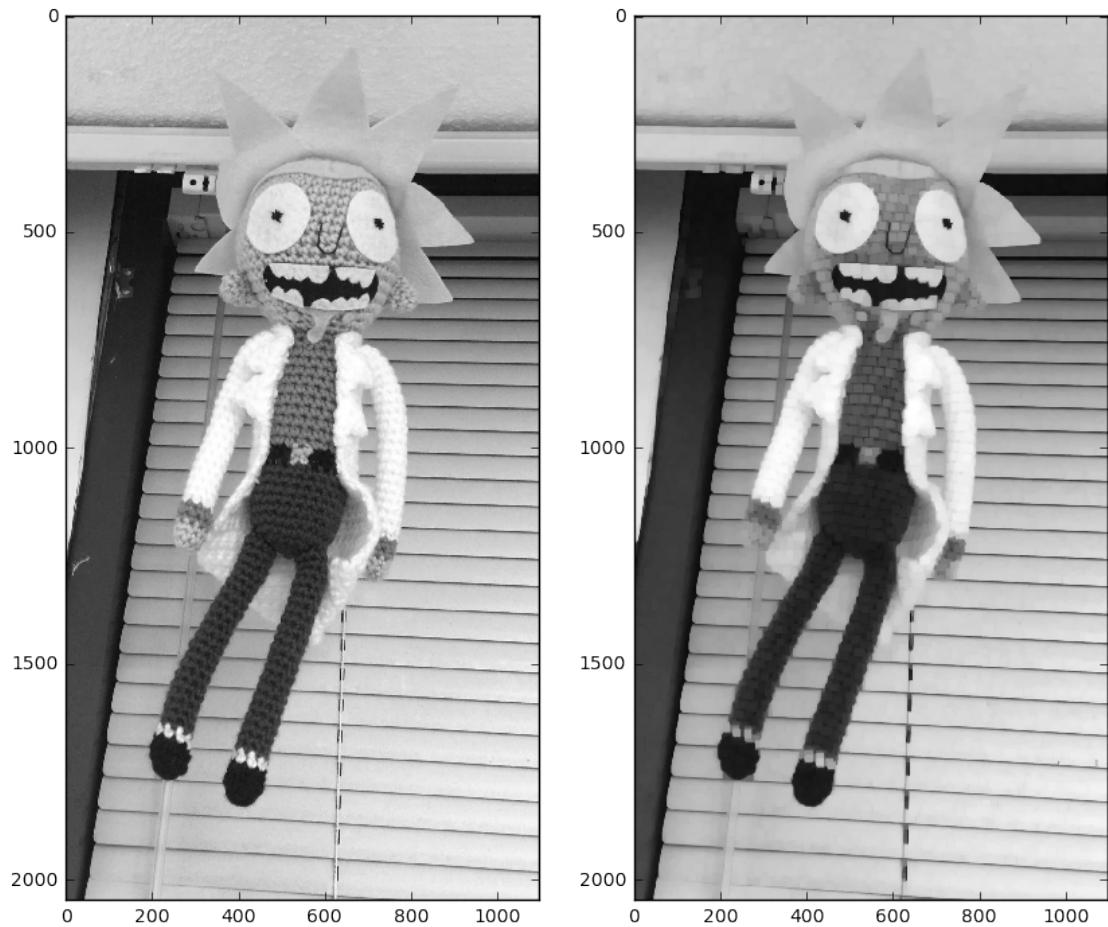


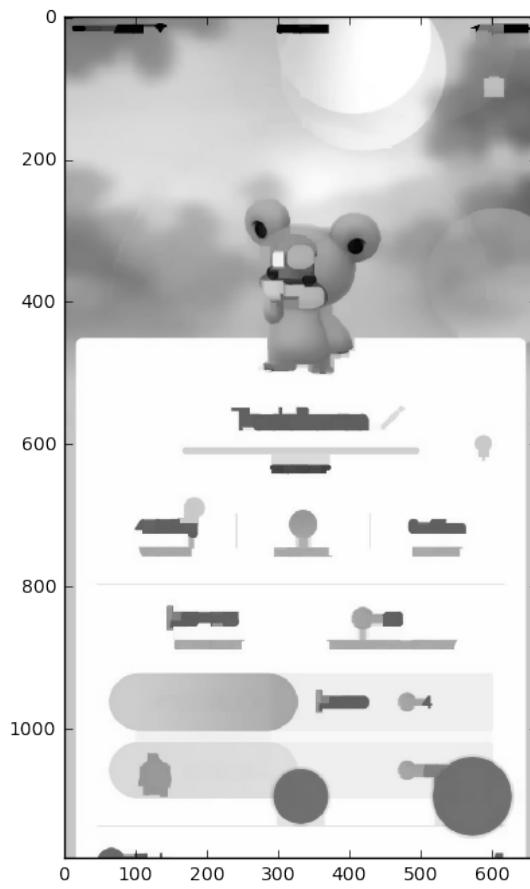
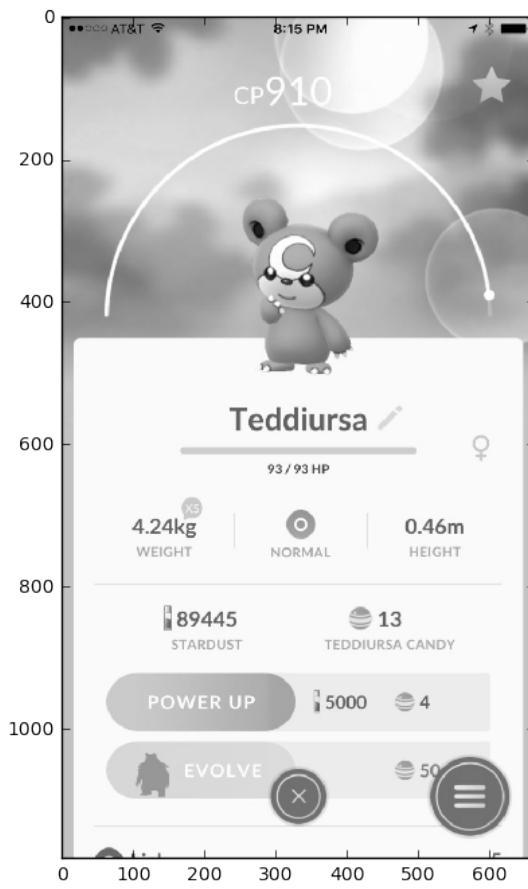
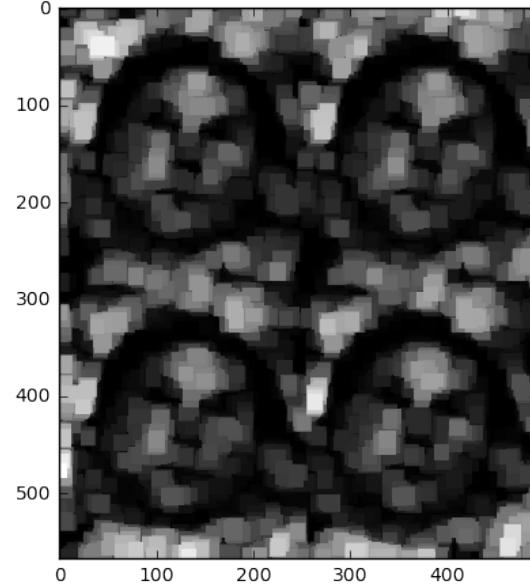
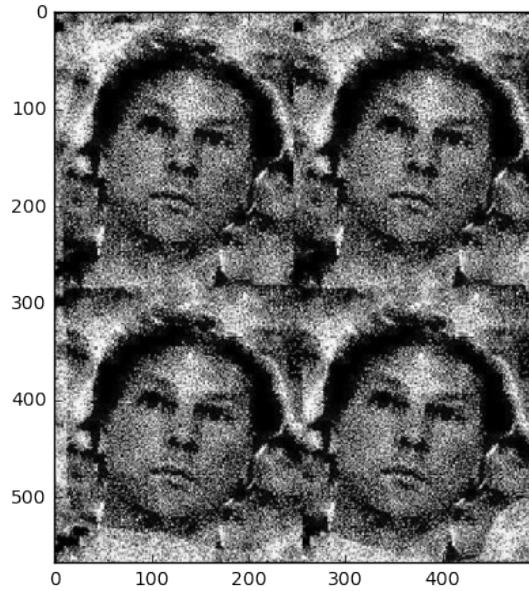


```
In [32]: # Apply to image set
size=15
for i,img in enumerate(imgset):
    imgbw = img_as_float(color.rgb2grey(img))
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.imshow(imgbw)
    plt.subplot(1, 2, 2)
    plt.imshow(morphological_denoise(imgbw, 'square', size))
```









In []:

In []:

2 Python Tutorials

Python 101 Beginning Python http://www.rexx.com/~dkuhlman/python_101/python_101.html

The Official Python Tutorial - <http://www.python.org/doc/current/tut/tut.html>

The Python Quick Reference - <http://rgruet.free.fr/PQR2.3.html>

YouTube Python Tutorials

Google Python Class - <http://www.youtube.com/watch?v=tKTZoB2Vjuk>

Python Fundamentals Training – Classes <http://www.youtube.com/watch?v=rKzZEtxIX14>

Python 2.7 Tutorial Derek Banas - http://www.youtube.com/watch?v=UQi-L-_chcc

Python Programming Tutorial thenewboston - <http://www.youtube.com/watch?v=4Mf0h3HphEA>

3 Evaluation

Install Anaconda 4 for Python 2.7 and get this notebook to run with a set of your images.