Histogram

Background

Image statistics can be calculated directly from the input image pixels. But there are times when the calculations are more efficient when performed on the image histogram. Furthermore, the histogram can provide visual cues as to the quality of the image. In this assignment you will create a histogram, display it, and calculate statistics from the histogram.

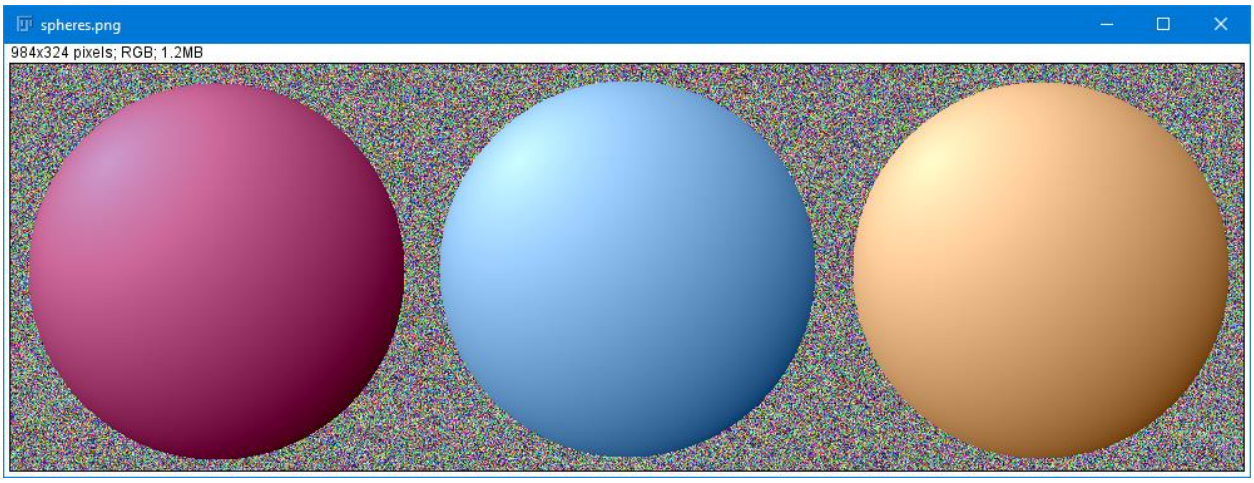
Assignment

1. Read the given image (PNG file, use **`ImageIO`** and **`BufferedImage`** classes)
2. Convert the **`BufferedImage`** to a 3D **`int`** array to prepare for processing
3. Compute the histograms on the red, green, and blue image planes
    a. Three separate histograms
4. Compute pixel value statistics from the red, green, and blue histograms and print values to the console similar to what is shown below
    a. Compute minimum, maximum, mean, standard deviation, median, mode

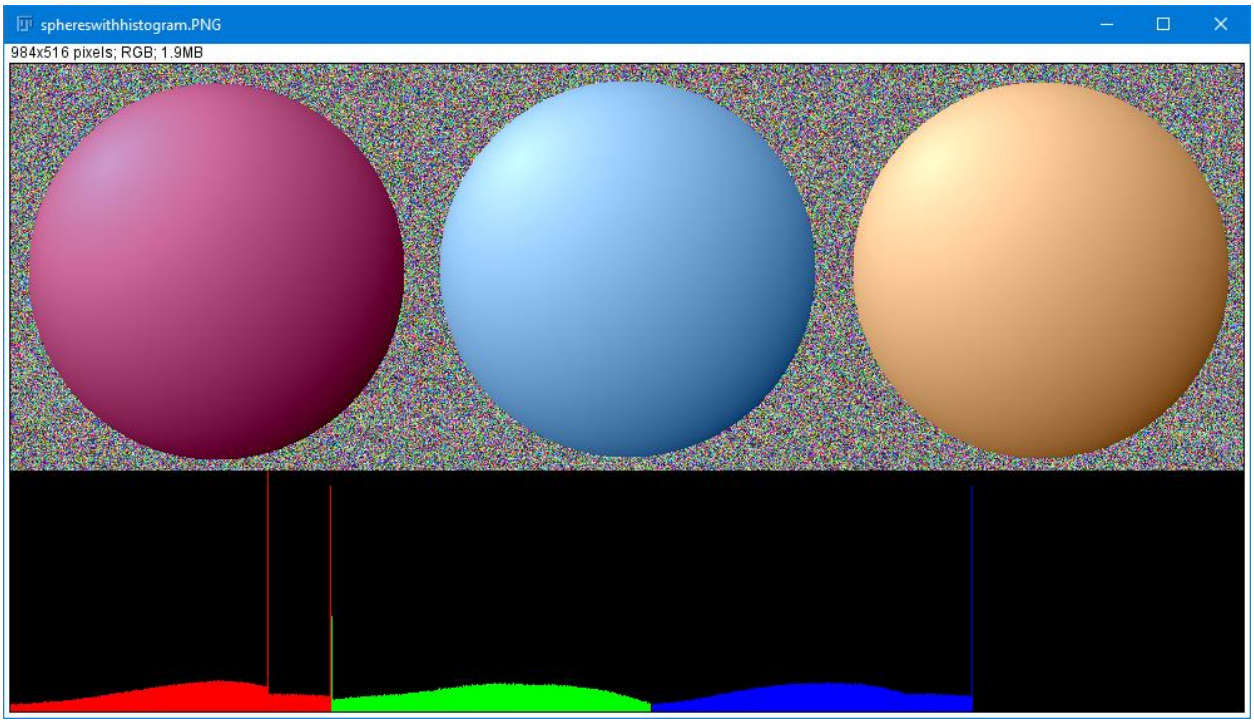
```
          min      max    mean     dev    median   mode
Red:    0.000  255.000 150.092 64.191  157.000 205.000
Grn:    0.000  254.000 129.605 66.377  134.000   0.000
Blu:    0.000  255.000 139.666 66.134  140.000 255.000
```

5. Create a new **`BufferedImage`** with the width of the input image and height of the input image plus 200 (*height + 200 x width*)
6. Place the input image into the **`BufferedImage`** at location (0, 0)
7. Scale and draw the red, green, and blue histograms into the **`BufferedImage`** below the input image
    a. Use the Graphics2D graphics context associated with the **`BufferedImage`**
8. Save the **`BufferedImage`** to a PNG file

The input image and resultant image with histograms are shown below.



Input Image



Input Image with Histograms added

Notes:

- You will need to scale the histograms for display. Use linear scaling, y = mx + b, with m = (200.0 / maximum value of all bins in all 3 histograms) (use the same scale factor for all 3 histograms) and b = 0. That is, calculate one maximum for all 3 histograms then compute

$$redhisto[bin] = \frac{192}{\max(all\ histogram\ bins)} * redhisto[bin]$$
$$greenhisto[bin] = \frac{192}{\max(all\ histogram\ bins)} * greenhisto[bin]$$
$$bluehisto[bin] = \frac{192}{\max(all\ histogram\ bins)} * bluehisto[bin]$$

  You will need to use **`double`** types for these calculations then cast to **`int`** as necessary.

Deliverables

1. Source code attached to assignment in Blackboard
    a. All .java files
2. A text document in PDF format (do not use any other format), that contains
    a. Reflective essay describing your experiences in writing this code including your degree of completion
    b. A screen shot of the pixel statistics as computed from the histograms (see above)
3. The PNG file of your image with histograms added (see above)