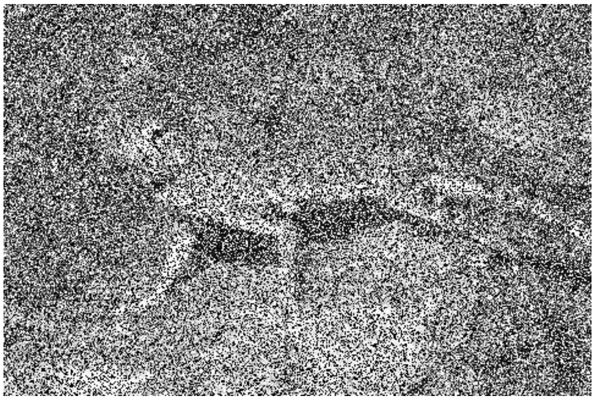
http://people.sc.fsu.edu/∼jburkardt/presentations/asa_2011_images_lab.pdf

Algorithms II - Lab 7:
ALGORITHMS FOR IMAGES



Problem 1: *Contrast Stretching*

Copy the file **mona_lisa.png** for this problem.

1. read in the image file, display the image;
2. try doubling the value of each pixel, and display the new image;
3. make a histogram of the image data, using 256 bins;
4. divide the range [0,255] into three ranges with about the same number of pixels (based on the histogram). Reset pixels in the first range to 0, the middle range to 127, the third range to 255. Display the 3-level image;
5. following the example on the slides, try to use the K-Means algorithm to choose K=8 representative gray levels. Reset the pixels to their representative values, and display the 8-level image.

Turn in 5 images, one for each step.

Problem 2: *Noise Removal*

Consider the following algorithms that try to improve a noisy image:

- Algorithm “Average” replaces each pixel by the average of the values in the 3x3 neighborhood.
- Algorithm “Median” replaces each pixel by the median of the values in the 3x3 neighborhood.

Write programs that implement both algorithms for a grayscale image. Apply each algorithm to the image *lizard_noisy*. Then apply each algorithm a second time, that is, to the ”improved image”. Does the image improve again?

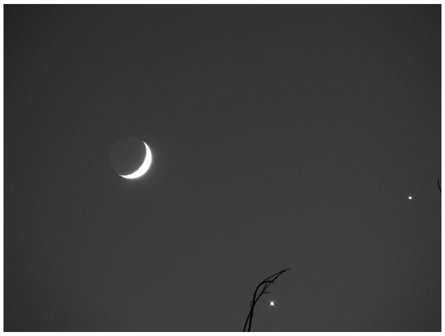
Turn in the four improved images.

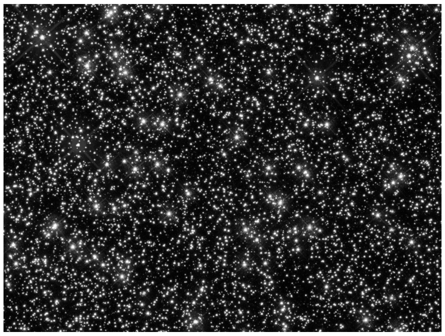
Problem 3: *Edge Detection*

Implement the NEWS algorithm for trying to detect edges in a grayscale image. (You are free to choose a more sophisticated algorithm if you like).

Apply your implementation of an edge detection algorithm to the *bird* image. Use some kind of thresholding (as we did in class) so that you can create a new picture which displays the locations of edges. Adjust the thresholding so that you get a "cartoon" outline of the bird.

Turn in the value of the threshold parameter, and your final image of the bird outline.





Problem 4: *Component Identification*

Get the *conjunction_gray* file.

Determine a threshold value **t** between 0 and 255 so that when you threshold the image, everything is black (0) but the two stars and the moon, which are white (255).

Apply a connected component identification algorithm to your thresholded image, which returns the number of components. If you have done things correctly, you should expect to get the answer "3".

Now work with the *star_field* image. This is going to be harder and you must use your judgment. First, threshold the image so that the black and white result seems to contain separated white blobs of stars. Then apply a component identification algorithm to get the number of components. How many stars does your algorithm "see"?

Turn in the threshold values and number of components for both images.

DATA FILES

The following files will be of use to you:

- **bird.png**, an image of a bird;
- **conjunction_gray.png**, an image of an astronomical conjunction;
- **lizard_noisy.png**, a noisy grayscale image of a lizard;
- **mona_lisa.png**, a dark grayscale image of the Mona Lisa;
- **star_field_gray.png**, an image of a field of stars;

These files will be available from the class's Blackboard site, or you can get them by going to
http://people.sc.fsu.edu/∼jburkardt/m_src/asa_2011_images_lab/asa_2011_images_lab.html