CSC 191B: Lab #7: Image Manipulation

Learning Outcomes

- Reading, visualizing, and writing image files in MATLAB
- Manipulating images for edge detection and blurring with Image Processing Toolbox functions

Background. Images are represented as 2D or 3D arrays, and there are three basic image types in MATLAB. In this lab we'll play around with all three. Basic MATLAB allows us to manipulate images in many ways, simply by treating them as arrays; MATLAB's Image Processing Toolbox has a long list of fancier algorithms for image manipulation. Write one script to complete the following tasks, and include answers to discussion questions in the comments of the script.

1 Clown

To load an example indexed image, type

```
load clown.mat
figure, imshow(X,map)
```

Note that `X` is matrix, and pull out the submatrix `X(100:140,30:80)`. Display the submatrix as an image, and then use `imwrite` to write it to a file called `clown-nose.png`.
Discussion question:

1.1. What is the index of the color (as given by the color `map`) that appears most frequently in the submatrix, and what are its RGB values? *Hint: MATLAB's `mode` will come in handy.*

2 Circuit

To load an example grayscale image, type

```
circuit = imread('circuit.tif');
figure, imshow(circuit)
```

Type `whos` or check your workspace to see what values are stored in the `circuit` array. An import image processing task is to detect certain features of images automatically. In this case, we'd like to detect the edges in the image – places where the luminosity changes drastically and quickly.

First convert these values to 0/1 (logical) values to get greater contrast by typing

```
circBW = imbinarize(circuit);
figure, imshow(circBW)
```

Then, to visualize edges in the image, type

```
figure, imshow(diff(circBW))
```

and write the edge image to a file called `circuit-edges.png`.

To see how well MATLAB's built-in function can do, try

```
figure, imshow(edge(circuit))
```

Discussion questions:

2.2. What does `diff(circBW)` compute and why are the dimensions different than `circBW`?

2.3. How is `edge(circuit)` better than the result from `diff(circBW)`?

3 Peppers

To load an example truecolor (or RGB) image, type

```
peppers = imread('peppers.png');
figure, imshow(peppers)
```

Instead of trying to highlight the edges of this image, we will try to smooth them out using a technique that is similar to how we simulated heat dissipation in Lab 6.

```
F = fspecial('average',7);
peppBlur = imfilter(peppers,F,'conv');
figure, imshow(peppBlur)
```

Note that `F` is just a 7×7 matrix with one value: 1/49. The `imfilter` algorithm with these arguments replaces each pixel value with average of the 49 surrounding pixels, which blurs the image. Convert the peppers image to grayscale, apply the blurring operation to the grayscale image, and the write the result to a file called `peppers-gray-blurred.png`.
Discussion questions:

3.4. Does the `edge` function have more trouble detecting edges in the blurred grayscale image than in the original grayscale image? Why?

3.5. What happens when you decrease or increase the value 7 in the use of `fspecial`?

What to turn in.

- One MATLAB script that to reads, manipulates, and writes images and includes the answers to the discussion questions.
- Three PNG files that contain the requested output images.

Grading rubric

- Code: 50 points for the script, which should be well-organized and well-documented
- Images: 10 points each for the output images
- Discussion: 4 points each for the discussion questions