CS 8 Lab: pa08

December 10, 2014

1 Overview

This assignment focuses on using the cImage.py functions. You'll make a ”pseudo-Warhol” image using functions that you write. This handout is a bit long, but if you follow it you should have no problem completing this assignment.

I'd like to take this opportunity to wish you a wonderful winter break! Thanks for being in my lab section. =)

2 Getting Set Up

Pull up the assignment description from `http://cs.ucsb.edu/~koc/cs8/hwexpa/pa08.html` and read through it. Download the `cImage.py` file found on the assignment description, and save it into the same folder where you'll save your `pa08.py` file.

Open up a new python file called `pa08.py` (use past handouts for help if needed). As the first three lines, type in comments containing your full name, lab section time, UCSB UMail email address, and perm number:

```
# Your name, Lab time
# Your UCSB email address
# Your perm number
```

In `pa08`.py, you have to write the following:

- Functions that manipulate pixels:
  - `MakeGrayPixel(p)`: takes the pixel p as input and returns a pixel q that is the grayscaled version of pixel p
  - `MakeNegativePixel(p)`: takes the pixel p as input and returns a pixel q that is the negative of pixel p
  - `IntensifyRedPixel(p, t)`: takes the pixel p and an integer t as input and returns a pixel q such that the intensity of the red component of pixel q is equal to $r \cdot t$, where r is the original red component (from 0 to 255). Note that the maximum value this should return is 255.
  - `IntensifyGreenPixel(p, t)`: like IntensifyRedPixel, but intensifies the green component of the pixel
  - `IntensifyBluePixel(p, t)`: like IntensifyRedPixel, but intensifies the blue component of the pixel
- Functions that manipulate images:

  - `MakeHalfImage(im1)`: takes the image $im1$ with an arbitrary size of $n \times m$ pixels, and returns $im2$ which is the same as $im1$, but of size $\frac{n}{2} \times \frac{m}{2}$ pixels
  - `MakeGrayImage(im1)`: takes the image $im1$ and returns a grayscaled-version of the image as $im2$
  - `MakeNegativeImage(im1)`: takes the image $im1$ and returns the negative version of the image as $im2$
  - `MakeIntenseImage(im1, t1, t2 t3)`: takes the image $im1$ and returns $im2$ such that the color intensity of each pixel from the original image becomes $r \cdot t1, g \cdot t2, b \cdot t3$ where r, g, b are the red, green, and blue components of the original pixels, respectively.

- The rest of your program: In the end, you want your code to read a rectangular image of size $n \times m$ pixels (where $n > m$) from an image file called ”input.gif,” and will create an image containing 9 sub-images, as described on the programming assignment. Please refer to `http://cs.ucsb.edu/~koc/cs8/hwexpa/pa08.html`

Let’s go ahead and set up our file for this. Don’t forget to import cImage! Everything will go in the same file, `pa08.py`:

```
from cImage import * #if you use this instead of import cImage, you don’t have to append cImage functio

##### pixel functions #####

def MakeGrayPixel(p):
    q = p # you’ll have to change this to the actual code
    return q

def MakeNegativePixel(p):
    q = p # you’ll have to change this to the actual code
    return q

def IntensifyRedPixel(p,t):
    q = p # you’ll have to change this to the actual code
    return q

def IntensifyGreenPixel(p,t):
    q = p # you’ll have to change this to the actual code
    return q

def IntensifyBluePixel(p,t):
    q = p # you’ll have to change this to the actual code
    return q

##### image functions #####

def MakeHalfImage(im1):
    im2 = im1 # you’ll have to change this to the actual code
    return im2

def MakeGrayImage(im1):
    im2 = im1 # you’ll have to change this to the actual code
    return im2

def MakeNegativeImage(im1):
    im2 = im1 # you'll have to change this to the actual code
    return im2

def MakeIntenseImage(im1,t1,t2,t3):
    im2 = im1 # you'll have to change this to the actual code
    return im2


##### rest of the program here: #####
```

Save your file. You should be able to run it with no errors at this point. Note that the lines like $p = q$ and $im2 = im1$ need to be deleted and replaced with the actual code. These are just in here so your code will run at this point.

3 Write your code

Now you want to fill this in with your code. Go function-by-function, and run your code each time you write a new function. You don't want to wait to run your code until you've written all of the functions. If you do, and there's an error, it will be much harder to figure out. Refer to the code examples from lecture on the Professor's website if you get stuck. Here are some suggestions for writing the code for each of the above functions:

- Tips for MakeGrayPixel:
  - `p.getRed()` returns the red value of the pixel p. Blue and green are similar.
  - A gray pixel will have the same value for red, green, and blue. You just need to figure out what this one value is.
  - Think: average.
  - Pixel(0, 1, 2) will return a pixel with a red value of 0, a green value of 1, and a blue value of 2.
- Tips for MakeNegativePixel:
  - Quite literally, think: negative.
  - The original red value of a pixel + the negative red value of a pixel should add up to 255.
- Tips for IntensifyRedPixel, IntensifyGreenPixel, IntensifyBluePixel:
  - Remember that the maximum value is 255. If you go over that, you need to just set it to 255. Something like this will work for IntensifyRedPixel:

    ```
    r = p.getRed()
        rnew = r*t
        if rnew>255:
           rnew = 255
    ```
- Tips for MakeHalfImage:
  - You can get the image width of $im1$ by saying `im1.getWidth()`. Similar for the height.
  - You make a new empty image called $im1$ of size $a \times b$ by saying `im2 = EmptyImage(a, b)`.
  - Create two loops, one that loops through the width in pixels of the new image, and one that loops through the height in pixels of the new image.

  - To calculate the r,g,b values of the new pixel, take the average of the four surrounding pixels from the original image.
  - To get the pixel in image at location i, j, type `image.getPixel(i,j)`.
  - To set the pixel in image at location i, j to be pixel p, type `image.setPixel(i,j,p)`.

- Tips for MakeHalfImage:
  - Create two loops, one that loops through the width in pixels of the new image, and one that loops through the height in pixels of the new image.
  - Use the function you already wrote for making gray pixels.
- Tips for MakeNegativeImage:
  - Create two loops, one that loops through the width in pixels of the new image, and one that loops through the height in pixels of the new image.
  - Use the function you already wrote for making negative pixels.
- Tips for MakeIntenseImage:
  - Create two loops, one that loops through the width in pixels of the new image, and one that loops through the height in pixels of the new image.
  - The code will be similar to the Intensify Color functions.

Awesome! Almost done! Now we have to write the main program, as per the assignment description. Here are a few tips:

- You should start with:

```
imf = FileImage("input.gif")
imx = MakeHalfImage(imf)
```

- You need to know the dimensions of the original image. Let these be n and m. Then type:

```
imy = EmptyImage(3*n//2,3*m//2)
```

- Use your functions to create images 1-9.
- Copy the pixels, one-by-one, from the images 1-9 into the new image, `imy`. Be sure that when you start copying a new image over that you start it at the correct location. You'll probably use loops, `getPixel()` and `setPixel()` to do this. Start with image1, then do image2, and so on.
- You might consider writing a function that copies pixels from an image to `imy` starting at a specific coordinate of `imy`, and then calling this multiple times.
- The last two lines of your code should be:

```
mywin = ImageWin("output",3*n//2,3*m//2)
imy.draw(mywin)
```

3.1 Running on the command line

If you're on a computer without IDLE or just want to run this on the command line, open up a Terminal window and navigate into the directory where you saved your pa08.py file. Type `python3 pa08.py` to run your program. It should create the new image. It may take a while, so be patient!

3.2 Running in IDLE

If you're on a computer with IDLE, go to `Run`→`Run Module` where you have your file. The IDLE prompt window (the window with the `>>>`) should restart. It should create the new image. It may take a while, so be patient!

4 Turnin

Ready to submit? Make sure you move your file over to CSIL first. **We only want you to turn in the python file. Do not turn in any screenshots.** Then, in a Terminal, navigate the the directory containing your file. To turn in, type the following command:

```
turnin pa08@cs8 pa08.py
```

and follow the on-screen directions. Remember, I will grade the last submission turned in before the deadline if you turn in multiple versions. **The deadline for this project is Friday, December 12th, 2014 at 11pm. We will not be accepting late submissions, so make sure you give yourself enough time to complete and turn in your project.**