ASSIGNMENTS

Assignment #6: Blending - Returned

Title	Assignment #6: Blending
Student	Tran, Ngoc T
Submitted Date	Oct 5, 2015 2:41 am
Grade	100.0 (max 100.0)

Instructions

Assignment 6: Blending

Introduction

In this homework assignment, we will be putting together a pyramid blending pipeline that will allow us to turn the following t
a blended image (This assignment will require you to create a blend using your own images!). This assignment is to be done ind
may ask others for help on Piazza, but you may NOT use other people's code outside of the code we have provided to you.

This is the output of blending the above two images with the mask.

Before we get started, Download the assignment files under Resources > Assignments > Assignment 6 that contains the files you
with.

Once you have extracted the file above, you are ready to get started.

Above you can see the sample images which are provided for you in the images/source/sample directory. Every time you run th
look inside each folder under images/, and attempt to find a folder that has images with filenames that contain 'white', 'black' a
it finds a folder that contains three images with those respective names, it will apply a blending procedure to them, and save t
images/output/. Along with the output image, it will create visualizations of the Gaussian and Laplacian pyramids used in the pı

Note: 'black' and 'white' are names to connect the images to the corresponding part of the black and white mask. The names do
any inherent properties of the images themselves, which are just two colored images.

The blending procedure takes two images and the mask and splits them into their red, green, and blue channels. It then blenc
separately. You do not have to worry about dealing with three channels, you can assume your images take in grayscale images (
always done).

always done).

The code will scale the mask image to the range [0,1], calculate the number of depth iterations, and construct a Gaussian pyra mask. It will then construct Gaussian pyramids and then Laplacian pyramids for the two images. Finally, it will blend the two La pyramids with the mask pyramid and collapse them to the output image.

Pixel values of 255 in the mask image are scaled to the value 1 in the mask pyramid, and assigned the strongest weight to the i 'white' during blending. Pixel values of 0 in the mask image are scaled to the value 0 in the mask pyramid, and assigned the str the image labeled 'black' during blending.

In order to facilitate this process, you will be providing six (6) key functions throughout this programming assignment that are th blocks of blending two images.

Part 0: Reduce and Expand functions.

See Module 04-03 on Pyramids.

As with previous assignments, running **assignment6_test.py** directly will apply a unit test to your code and print out helpful fee use this to debug your functions.

reduce

This function takes an image, convolves it, and then subsamples it down to a quarter of the size (dividing the height and width that we say subsample here. We recommend you look into numpy indexing techniques to accomplish the subsampling (essential index every other row and every other column).

Within the code, you are provided with a generating_kernel(a) function. This function takes a floating point number a, and retu generating kernel. For the reduce and expand functions, you should use **a = 0.4.**

Seeing as we have already covered how convolve works in class, for this assignment we allow you to use the scipy library imple convolve.

```
scipy.signal.convolve2d(image, kernel, 'same')
```

This call will convolve the image and kernel, and return an array of the 'same' size as image.

expand

This function takes an image and supersamples it to four times the size (multiplying the height and width by two (2)). After inc we have to interpolate the missing values by using the same convolution we used in reduce, and lastly we scale our output by

Some tips in terms of how to super sample an image.

1. Create an image that is twice the size of the input.
2. Review your reduce code. Look at what you did to get your output.
3. Instead of choosing every other row / col in reduce for your output, can you assign every other row / col in expand for y
4. As stressed above, look into numpy indexing. A key thing to note is the basic slice syntax, think about how you can use t advantage.

For this part of the assignment, please use the generating kernel with **a = 0.4** and the convolve2d function from the reduce fur
above.

Part 1: Gaussian and Laplacian Pyramids

In this part of the assignment, you will be implementing functions that create Gaussian and Laplacian pyramids. As usual, use **assignment6_test.py** to test your code. In addition, assignment6_test.py defines the functions viz_gauss_pyramid and viz_lapl_p take a pyramid as input and return an image visualizaiton. You might want to use these functions to visualize your pyramids whi we use these at the end of the testing to output your results!

gaussPyramid

This function takes an image and builds a pyramid out of it. The first layer of this pyramid is the original image, and each subsec the pyramid is the reduced form of the previous layer. Put simply, you are iteratively calling the reduce function on the outpu call, with the first call simply being the input image.

Please use the reduce function that you implemented in the previous part in order to write this function.

laplPyramid

This function takes a Gaussian pyramid constructed by the previous function, and turns it into a Laplacian pyramid. The code do contains further information about the operations you should perform for each layer.

Like with Gaussian pyramids, Laplacian pyramids are represented as lists of numpy arrays in the code.

Please use the expand function that you implemented in the previous part of the code in order to write this function.

Part 2: Writing the blend and collapse functions.

In this part, you will be completing the pipeline by writing the actual blend function, and creating a collapse function that will convert our Laplacian pyramid into an output image.

As always, you can use **assignment6_test.py** to test your code.

blend

This function takes three pyramids:

- white - a Laplacian pyramid of an image
- black - a Laplacian pyramid of another imge.
- mask - a Gaussian pyramid of a mask image.

It should perform an alpha-blend of the two Laplacian pyramids according to the mask pyramid. So you will be blending each pa together using the mask of that layer as the weight.

As described in the code documentation, pixels where the mask is 1 should be taken from the white image, pixels where the m be taken from the black image. Pixels with value 0.5 in the mask should be an equal blend of the white and black images. This mathematically described in the assignment comments.

You may assume that all of the provided pyramids are of the same dimensons, and have dtype float. You may further assume th pyramid has values in the range [0,1].

Your output pyramid should be of the same dimensions as all the inputs, and dtype float.

collapse

This function is given a Laplacian pyramid and is expected to 'flatten' it to an image.

We need to take the bottom (smallest) layer, expand it, and then add it to the next layer. We continue this process until we re the pyramid. Once you add the second layer to the top layer, the top layer is your output (a common mistake is to re-add the t top layer so if your values look like they are twice as big as what the expected output is, that is what you are doing).

This function should return a numpy array of the same shape as the top (largest) layer of the pyramid, and dtype float.

The Writeup

This is what we want you to do for the PDF.

1. Choose a 'black' image and a 'white' image that you would like to blend. Note: These do not have to be grayscale images light images. Our samples are two colored images.
2. Choose a unique mask (don't use the one we provide) that you created and explain how you created it.
3. Demonstrate these three images in the PDF.
4. Explain what you did to tackle each function. Note: For the expand function, we also want you to explicitly state why yo your output by 4.
5. Anything else you wish to note, feel free to include the pyramid outputs if you found them interesting, etc (for Peer Fee

What to turn in:

- **assignment6.py** - Your code.
- **assignment6.pdf** - See above for the writeup, don't forget to include your images! As stated in previous assignments, kee size of your PDF to 6MB, feel free to use www.smallpdf.com/compress-pdf to compress a larger file.

Submitted Attachments

assignment6.py (10 KB; Oct 4, 2015 11:55 pm)
assignment6.pdf (2 MB; Oct 5, 2015 2:41 am)

Additional instructor's comments about your submission

100/100. Great job! Good detail for the writeup.
Programming: 80/80
Writeup: 20/20

Demonstrate the 3 images in the PDF: 3/3
Explain each of the functions: 12/12
- reduce 2/2
- expand 2/2
- gaussPyramid 2/2

- laplPyramid 2/2
- blend 2/2
- collapse 2/2

Use your own mask: 5/5

Instructor's attachments to this submission

feedback.txt (4 KB; Oct 19, 2015 12:15 pm)