CSC 589 Homework 3: Image Blending

Brief

Due date: March 8, Sunday, end of day
Team work: encouraged, maximum number of members, 2 (This is restricted to two people per team)
Totals points: 100pts + 20pts

Overview

The goal of this project is to use image-blending techniques to seamlessly merge information from multiple images. The primary focus of this project is to have fun and experiment. You will implement Laplacian blending, a reasonably simple technique. If you are interested in pursuing this further, consider doing Poisson image blending as a final project (Perez 2003). Please talk with me about this if you are interested in perusing this as your final project. Both the original Gaussian pyramid paper and the Poisson image blending paper are attached in the homework folder for your references.

Details

Begin by acquiring images. You can acquire images from your phones and cameras, as well as download from the Internet. The apple and orange images are provided in the data folder. However, it is required you use additional images for this project. I also provided once again my GeneratePyramid.py code that we have been playing around in class.

Laplacian Pyramids. Implement functions that build and collapse Laplacian pyramids. The input to the Laplacian pyramid building function is an image and the output is both the Gaussian and Laplacian pyramids for the image. The situation is reversed for collapsing a Laplacian pyramid (but really all that is needed is the lowest level Gaussian along with all levels of the Laplacian pyramid). You can use the *interpolate* and *decimate* functions discussed in class, or you can implement your own. Note that building and collapsing a Laplacian pyramid should yield the original input image. Before you do any blending, checking whether your code can successfully collapsing into multiple sub-bands and reconstruct an image.

Create an image that shows multiple levels of the Gaussian and Laplacian pyramid of an image and include this in your write up along with an explanation of image pyramids in your own words.

Laplacian Blending. Implement Laplacian image blending for a pair of images and a binary mask. In Laplacian blending, a new Laplacian pyramid is built by alpha

blending each corresponding Laplacian pyramid level for the input image pairs using a Gaussian pyramid of the binary mask as the alpha mask. The lowest level of the new pyramid also includes the alpha blend of the lowest level Gaussian images in the two pyramids (you can blend all levels of the Gaussian pyramids if you want, but you technically only need the coarsest level). The resulting Laplacian pyramid is collapsed to recover the blended image.

Try your multi-scale Laplacian blending method out on several sets of images. Show *at least three* results you consider "good" and one you consider "poor" in your write up. For each image, discuss why you think this approach did or did not work well.

You can compare your results with a simple “copy and paste” methods by directly apply mask on one image and blend with another.

Submit your final code onto GitHub.

Tips

To blend a pair of images you need a source image, target image, and mask. The source image contains the data you are inserting into the target image. The mask indicates where in the output image the data comes from the source image. It helps if all three images are the same size. A simple mask can be built by dividing the image plane in half. This will blend two images together along the middle seam, which is what is done for the famous apple/orange hybrid.

More advanced masks can be created using an image editing program that uses layers, such as Gimp. A simple procedure is:

- Open the target image
- Open the source image
- Select a region in the source image (rectangle, oval, lasso). Copy it.
- Paste the region into the target image as a new layer.
- Move or rescale the new layer so it appears where you want it to above the target layer.
- Create a new transparent layer.
- Paint in this layer with white paint where you want the source image data to appear in the final image.
- For each layer in turn, make the layer exclusively visible, select Save As... and save the file with a unique name, like myblend-source.png.

Note that Gimp on the Mac is a little annoying because you have to click in a window to change focus before you can interact with the window.

Valid mask values used for alpha blending are between zero and one.

$$I_o(x,y) = \alpha(x,y)I_s(x,y) + (1-\alpha(x,y))I_t(x,y), \quad \alpha \in [0 \ldots 1]$$

If you load an alpha mask from a file, you will probably need to modify the image data to fit within this range. You may also want to make sure you are using the same mask for all color channels.

Bells & Whistles (Extra Points)

- (5pts) Be exceptional creative of acquiring and your choices of source and target images.
- (5pts) You can discuss the shortcomings you discovered with the Pyramid method. When does the method fail?
- (10pts) Implement the two-band Laplacian blending described in Brown and Lowe, Recognizing Panaroams, ICCV 2003. You can simply jumpt to section 5, last two paragraphs. Compare with the multi-band method you have already have implemented.

Web-Publishing Results

All the results for each project will be put on the course website so that the students can see each other's results. The professor will select "winning" projects that impress us and there will be in class presentations for these projects. If you do not want your results published to the web, you can choose to opt out. If you want to opt out, email me.

Write up

For this project, and all other projects for the rest of the semester, you must do a project report in HTML. We provide you with a placeholder .html document, which you can edit. In the report you will describe your algorithm and any decisions you made to write your algorithm a particular way. Then you will show and discuss the results of your algorithm. In the case of this project, show the results of your blending algorithm and show the good and bad examples. Also, discuss anything extra you did. Feel free to add any other information you feel is relevant. If you performed this assignment as a team, please indicate this in the start of your webpage and credit each person for his/her contributions.

Please do not move the highlighting folder that contains important CSS sheets (unless you want to do your own web design). Just edit the .html file and keep the folder structure when you submit your code.

The details of how to setup your webpage on the AU server can be found here:

http://www.american.edu/oit/network/WebPage-Students.cfm

But you don't have to have Internet to view the webpage.

Handing in

This is very important, as you will lose points if you do not follow instructions. Every time after the first that you do not follow instructions, you will lose 5 points. The folder you hand in must contain the following:

- README - text file containing anything about the project that you want to tell the TAs
- code/ - directory containing all your code for this assignment
- html/ - directory containing all your html report for this assignment (including images). Only this folder will be published to the course web page, so your webpage cannot contain pointers to images in other folders of your hand-in.
- html/index.html - home page for your results
- Zip the folder and submit to blackboard.
- You can upload your folder onto your own website. If you do so, you are still required to submit the folder and the address of the website via blackboard.

Rubric

- +40 pts: Working implementation of Laplacian pyramid decomposition and image reconstruction.
- +30 pts: Successfully blending images using the algorithm.
- +20 pts: Writeup with several examples of blended images and discuss successful and failed examples.
- +20 pts: Extra credit. See the above for details.
- -5*n pts: Lose 5 points for every time (after the first) you do not follow the instructions for the hand in format
- Late policy applies! Notice project will not be accepted after 3 days.

There are plenty of codes with Python on Pyramid image blending. Please DO NOT copy the code form the internet since I have a good grasp of how they look like. This assignment was modified from multiple resources (Colby college, CMU, etc). But web plagiarism is strictly forbidden.