CSCE 448/748 - Computational Photography

Programming Assignment 3

Deadline: Feb. 26th

1 Goal

The goal of the first part of the project is to create hybrid images using the approach described in the SIGGRAPH 2006 paper by Oliva, Torralba, and Schyns. Hybrid images are static images that change in interpretation as a function of the viewing distance. The basic idea is that high frequency tends to dominate perception when it is available, but, at a distance, only the low frequency (smooth) part of the signal can be seen. By blending the high frequency portion of one image with the low-frequency portion of another, you get a hybrid image that leads to different interpretations at different distances.

The goal of the second part of this project is to perform image blending using pyramids. Basically, given a source image, a target image, and a mask, our goal is to blend the masked areas of the source image into a target photograph in a seamless manner.

2 Starter Code

Starter code (in Python) along with the images can be downloaded from here. Note that, expected result for hybrid images is included in the "Results" folder.

3 Task 1

The starter code takes the correspondences between the two input images to align them. The alignment is important because it affects the perceptual grouping (read the paper for details).

First, you'll need to get a few pairs of images that you want to make into hybrid images. You can use the sample images for debugging, but you should use your own images in your results. Then, you will need to write code to low-pass filter one image, high-pass filter the second image. For a low-pass filter, Oliva et al. suggest using a standard 2D Gaussian filter. For a high-pass filter, they suggest using the impulse filter minus the Gaussian filter (which can be computed by subtracting the Gaussian-filtered image from the original). The standard deviation of each filter should be chosen with some experimentation. Note that you need to write your own code to generate the Gaussian kernel by approximating the continous Gaussian function as done in the class. Once you create your kernel, filtering can be done by `scipy.signal.convolve2d`.

Show your results on two examples in addition to the one provided.

3.1 Extra Credit

Try using color to enhance the effect. Does it work better to use color for the high-frequency component, the low-frequency component, or both? Discuss it in the report.

4 Task 2

Here, you implement the pyramid blending approach that was discussed in the class. Given a source, target, and a mask, you have to follow the following steps to generate the blended image:

- Compute Laplacian pyramids L_S and L_T from the source and target images.
- Compute a Gaussian pyramid G_M from the mask.
- Use the the Gaussian pyramid to combine the source and target Laplacian pyramids as follows:

$$L_C(l) = G_M(l) \times L_S(l) + (1 - G_M(l)) \times L_T(l), \tag{1}$$

where l is the layer index.

- Collapse the blended pyramid L_C to reconsruct the final blended image

You can test your implementation on image 1. You should also create one example of your own where pyramid blending produces a nice blended image. Additionally show an example where the approach fails (explain why it fails). You can use the `GetMask` function to obtain a mask on your images.

You should use `skimage.transform.resize` to build the Gaussian and Laplacian pyramids. Note that `skimage.transform.resize` already performs the filtering and subsampling. So you don't need to worry about aliasing. Just use this function to implement the Reduce and Expand operations.

5 Write Up

Include all the results in your report. For hybrid images, show the inputs as well as the hybrid image. For pyramid blending, you should show the input images, as well as the naive blending along with your blended result. Describe how you implemented the assignment. Discuss any problem you faced when implementing the assignment or any decisions you had to make.

6 Graduate Credit

Graduate students have to do the extra credit.

7 Deliverables

Your entire project should be in a folder called "`firstname_lastname`". This folder should be zipped up and submitted through e-campus. Inside the folder, you should have the followings:

- A folder named "Code" containing all the codes for this assignment. Please include a README file to explain what each file does if you add any other files to the starter code.
- A report in the pdf format. **Make sure you write your name on top of the report. Also make sure the pdf file is under 5 MB.**

Make sure you exclude all the results and original images from your submission.

8 Checklist

Make sure you can check all the items below before submitting your assignment. You will lose 5 points for each item that cannot be checked.

☐ The folder is named properly ("`firstname_lastname`"). Note _ between first and last name. The folder structure should be exactly as follows

"`firstname_lastname`.zip"

- "`firstname_lastname`"
  - "Code"
  - Report.pdf

☐ Inside the root folder, there is a folder called "Code" that contains your source code. Also make sure the report is in the root folder.

☐ The folders "Images" and "Results" are not included (you only submit your codes and a report).

☐ Name written on top of the report.

☐ The report is in pdf format and the file is under 10 MB.

☐ All the results are included in the report.

☐ Original images or results are not included in the package.

9 Ruberic

Total credit: [100 points]

[30 points] - Implement hybrid images
[10 points] - Include two example of your own
[40 points] - Implement pyramid blending
[10 points] - Include two example of your own (success and failure)
[10 points] - Write up

Extra credit: [5 points]

[5 points] - Implement hybrid images in color

10 Acknowlegements

The project is derived from Alyosha Efros Computational Photography course with permission.