National Tsing Hua University
Department of Electrical Engineering
EE3662 Digital Signal Processing Laboratory, Fall 2022

Lab 10 Image Filtering and Hybrid Images

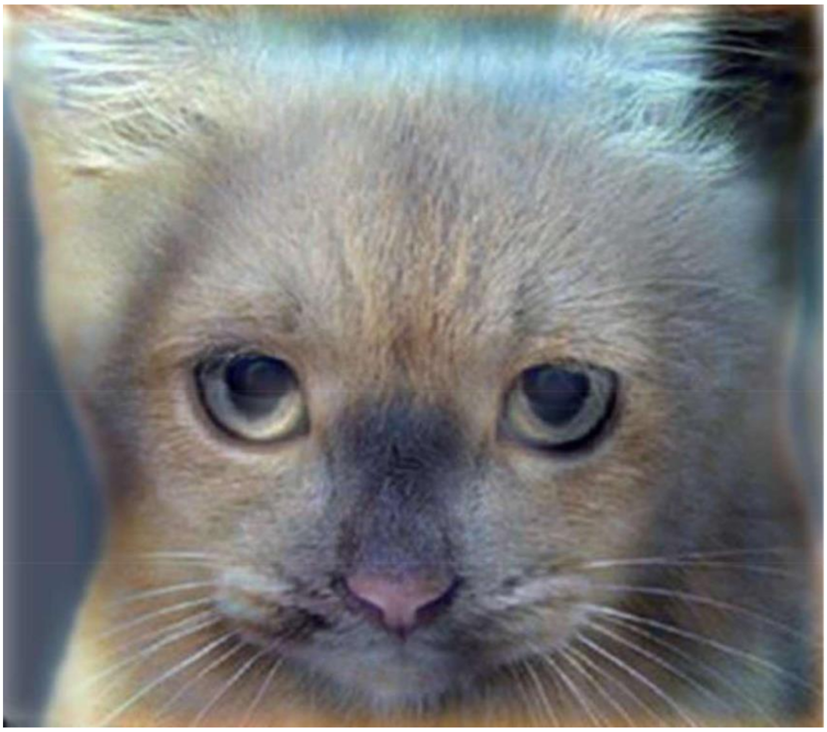
Assigned on Nov 28, 2022

Due by Dec 5, 2022

Overview

The goal of this assignment is to write an image filtering function and use it to create hybrid images using a simplified version of [1].

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. However, at a distance, only the low frequency (smooth) parts of the signal can be seen. By blending the high frequency portion of one image with the low frequency portion of another, you can get a hybrid image that leads to different interpretations at different distances.



In-class Demo

1. Image Filtering (30%)

Image filtering (or convolution) is a fundamental image processing tool. See chapter 3.2 of Szeliski and the lecture materials to learn more about image filtering. MATLAB has many built-in and efficient functions to perform image filtering, but you should write your own such function in this assignment.

You need to implement `my_imfilter()` which imitates the default behavior of the build- in `imfilter()` function. As specified in `my_imfilter.m`, your filtering algorithm must:

(1) Support grayscale and color (RGB) images.

(2) Support arbitrary shaped filters, as long as both dimensions are odd (e.g. 7x9, 3x21, or 11x11). This restriction makes the location of center pixel in filters clear.

(3) Pad the input image with zeros, and return a filtered image which has the same resolution as the input image.

We have provided a script, `test_my_imfilter.m`, to help you debug your image filtering algorithm.

Potentially useful MATLAB functions: `padarray()`

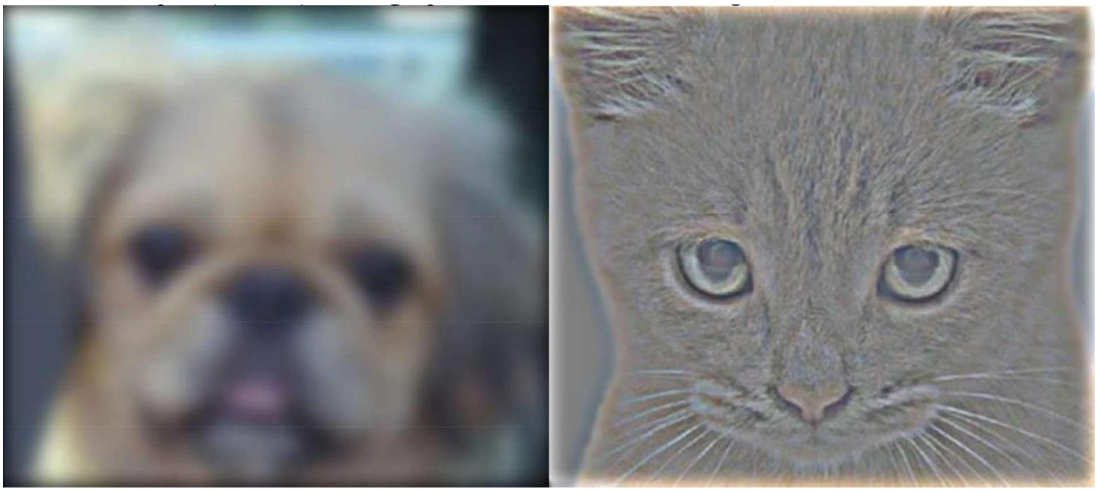
Forbidden functions: Built-in filtering functions in general. E.g. `Imfilter(), filter2(), conv2(), nlfilter(), colfilt()` (You can use them for testing but not in your final code.)

2. Hybrid Image (20%)

A hybrid image is the sum of a low-pass filtered version of one image and a high-pass filtered version of another image. There is a hand-crafted parameter called the “***cutoff frequency***”, which can be tuned for each image pair. It controls the amount of high frequency signal to be removed from the first image and also the amount of low frequency signal to be removed from the second image. In the paper, it is suggested to use two cutoff frequencies (one tuned for each image) and you are free to try that as well. In the starter code (`generate_hybrid_image.m`), the cutoff frequency is controlled by changing the standard deviation of the Gaussian filter used in constructing the hybrid images.

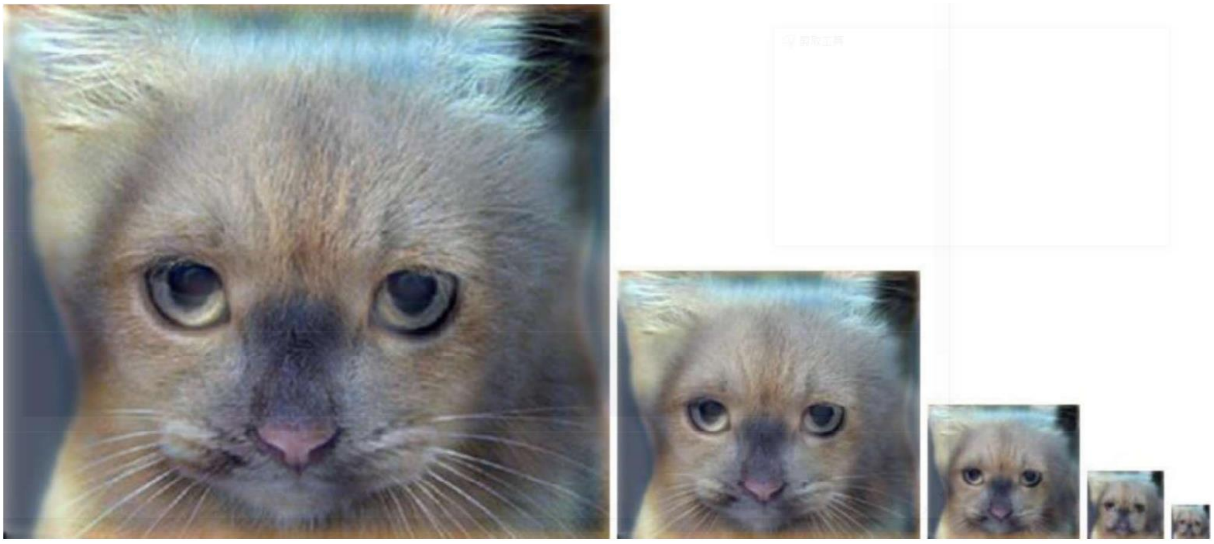
For the hybrid image shown in the overview part, the two source images look like this:

The low-pass (blurred) and high-pass versions of these images look like this:



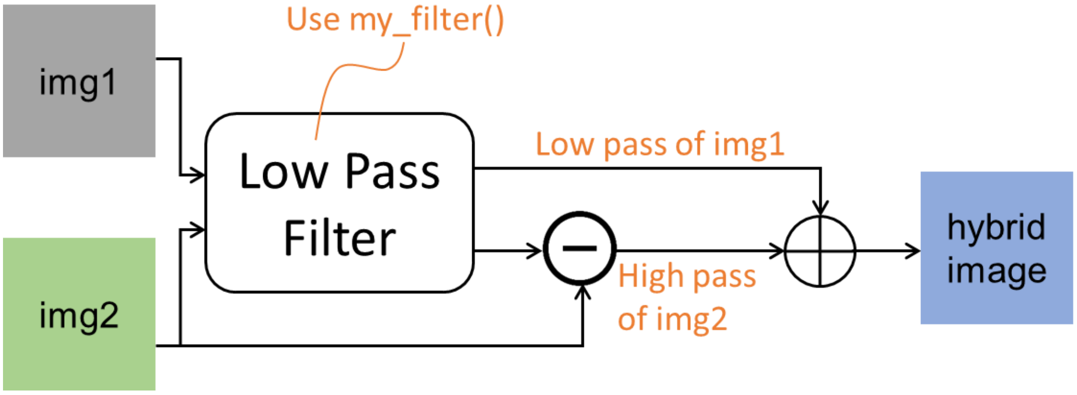
The high frequency image is actually zero-mean with negative values. Hence it is visualized by adding 0.5 to its values. In the filtered image, bright pixels are positive and dark pixels are negative.

Adding the high frequency and low frequency images together gives you the hybrid image shown in the overview part. If you're having trouble seeing the multiple interpretations of the image, a useful way to visualize the effect is by progressively downscale the hybrid image as below:



The starter code provides a function (**`vis_hybrid_image.m`**) to save and display such visualization.

Overall the processing flow to generate hybrid image is shown below.



Report

1. (6%) Explain why the high frequency images will be seen at close distance (larger one) while the low frequency images will be seen at far distance (smaller one)? (Hint: What's the meaning of high frequency signal and low frequency signal in an image?)
2. (9%) Discuss different low pass filters (box filter and Gaussian filter under the same size). Which one generates visually-pleasing hybrid images? Please show some examples to explain. (You are free to use the images provided in Lab10/data/). A size of 3x3 box filter is shown below.

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

3. (10%) Reverse the flow when generating low-passed filtered image and high-passed filtered image. Specifically, low pass filtered image is obtained by subtracting high pass filtered image. High pass filter is provided below. (Put your code in Lab10/code/ and name it **`reverse_generate_hybrid_image.m`**) Please discuss the difference between this reversed flow and flow with Gaussian filter used in this lab. (e.g. quality of hybrid image and low-pass filtered image) (You are free to use the images provided in Lab10/data/)

$$\begin{bmatrix}0 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & 0\end{bmatrix}$$

4. (7%) Try different settings of the cutoff frequency (e.g. 3, 7, 11…) in **`generate_hybrid_image.m`**. Show your results and explain what you observe (e.g. differences between the results of different cutoff frequencies, execution time, filter kernel size, blurred quality…).
5. (15%) Alignment of two images before blending them together is important in implementing hybrid images. Try to create your own alignment flow for two arbitrary images. **These two images should be in different scales. The sizes and the locations of the objects in the images**

that you want to blend should also be different. Please specify the steps and details in your flow. Implement hybrid images with your own flow and show the results. Please put your source images in Lab10/data/ and your results in Lab10/results/. If you have any code, please put it in Lab10/code/. (Note: For this question, the larger portion of your flow implemented with coding, the higher score you will get.)

6. (3%) Conclusion.

Deliverable and file organization

Directory	Filename	Description
Lab10/code/	*.m	All matlab codes
Lab10/data/	*.png / *.jpg	Your own source images
Lab10/results/	*.png / *.jpg	Your results
Lab10/report/	report_10xxxxxxx.pdf	Your report

Please organize your files according to the above table, and change the root name from Lab10 to Lab10_10xxxxxxx. Then compress it as Lab10_10xxxxxxx.zip in ZIP format. (Note: 10xxxxxx is your student ID)

Lab10
Lab10_10xxxxxxx
Lab10_10xxxxxxx.zip

Wrong file delivery, wrong file organization or wrong file naming will get up to 5% punishment.

Reference

[1] Aude Oliva, Antonio Torralba, and Philippe G. Schyns. 2006. Hybrid Images. ACM Trans. Graph. 25, 3 (July 2006), 527–532. DOI:https://doi.org/10.1145/1141911.1141919