



(Look at image on right from very close, then from far away.)

Project 1: Image Filtering and Hybrid Images

Overview

The goal of this assignment is to create hybrid images using a modification of 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.

Details

This project is intended to familiarize you with MATLAB and image filtering. Once you have created an image filtering function, it is relatively straightforward to construct hybrid images. If you don't already know MATLAB, you will find this [tutorial on MATLAB](#) helpful.

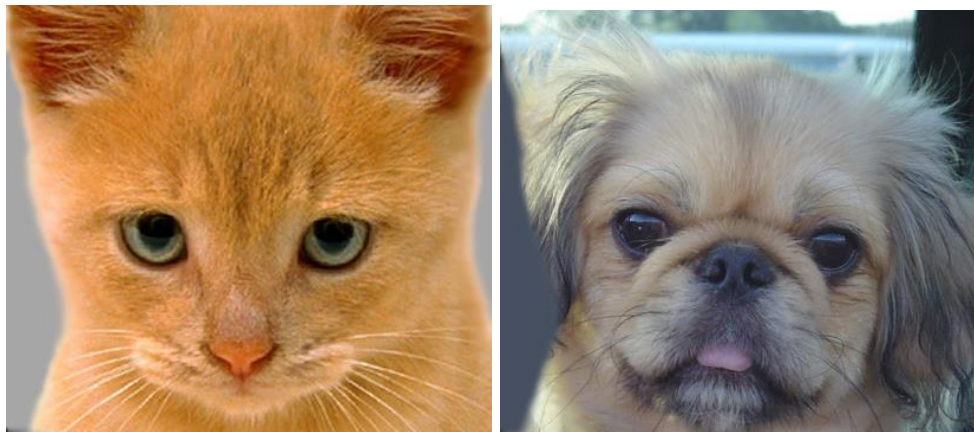
Image Filtering. Image filtering (or convolution) is a fundamental image processing tool. See chapter 3.2 of Szeliski and the lecture materials to learn about image filtering (specifically linear filtering). MATLAB has numerous built in and efficient functions to perform image filtering, but you will be writing your own such function from scratch for this assignment. More specifically, you will implement *my_imfilter()* which imitates the default behavior of the built in *imfilter()* function. As specified in *my_imfilter.m*, your filtering algorithm must (1) support grayscale and color images (2) support arbitrary shaped filters, as long as both dimensions are odd (e.g. 7x9 filters but not 4x5 filters) (3) pad the input image with zeros or reflected image content and (4) return a filtered image which is the same resolution as the input image. We have provided a script, *proj1_test_filtering.m*,

to help you debug your image filtering algorithm.

Hybrid Images. A hybrid image is the sum of a low-pass filtered version of the one image and a high-pass filtered version of a second image. There is a free parameter, which can be tuned for each image pair, which controls how much high frequency to remove from the first image and how much low frequency to leave in the second image. This is called the "cutoff-frequency". 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, the cutoff frequency is controlled by changing the standard deviation of the Gaussian filter used in constructing the hybrid images.

We provide you with 5 pairs of aligned images which can be merged reasonably well into hybrid images. The alignment is important because it affects the perceptual grouping (read the paper for details). We encourage you to create additional examples (e.g. change of expression, morph between different objects, change over time, etc.). See the [hybrid images project page](#) for some inspiration.

For the example shown at the top of the page, the two original 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 so it is visualized by adding 0.5. In the resulting visualization, bright values are positive and dark values are negative.

Adding the high and low frequencies together gives you the image at the top of this page. If you're having trouble seeing the multiple interpretations of the image, a useful way to visualize the effect is by progressively downsampling the hybrid image as is done below:



The starter code provides a function `vis_hybrid_image.m` to save and display such visualizations.

Potentially useful MATLAB functions: `fspecial()` and the operators in the MATLAB tutorial which make it efficient to cut out image subwindows and do the convolution (dot product) between them. `padarray()`.

Forbidden functions you can use for testing, but not in your final code: `imfilter()`, `filter2()`, `conv2()`, `nlfilter()`, `colfilt()`.

Writeup

For this project, and all other projects, you must do a project report. 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 filtering algorithm (the test script saves such images already) and show some of the intermediate images in the hybrid image pipeline (e.g. the low and high frequency images, which the starter code already saves for you). Also, discuss anything extra you did. Feel free to add any other information you feel is relevant.