TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

CS418: COMPUTER VISION

Instructor: Effrosyni Doutsi

Assignment 1: Filtering and hybrid images Due date: Sunday, April 18, 2021

This assignment is adapted from Svetlana Lazebnik

The goal of this assignment is to write an image filtering function and use it to create hybrid images. 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 image filtering. Once you have created an image filtering function, it is relatively straightforward to construct hybrid images.

Image filtering (or convolution) is a fundamental image-processing tool. 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 MATLAB 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.

You are provided a script, assignment1_filtering_test.m, to help you debug your image filtering algorithm.

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 relevant 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.

You are provided with 2 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 example, two possible original images might 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 visualize_hybrid_image.m to save and display such visualizations. Potentially useful MATLAB functions: fspecial() and the operators which make it efficient to cut out image subwindows and do the convolution (dot product) between them. padarray() does zero padding for you.

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

What to turn in:

You should turn in both your **code** and a **report** discussing your solution and results.

The report should contain a description of 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. 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). A good writeup doesn't just show results, it tries to draw some conclusions from your experiments.

Turning in the Assignment

Your submission should consist of the following:

- All your code and output images **in a single zip file**.
- A brief report **in a single PDF file** with all your results and discussion.

Academic integrity: Feel free to discuss the assignment with each other in general terms. Coding should be done a product of each individual team. If you use existing material, be sure to acknowledge the sources in your report.