Introduction to Computer Vision Assignment #1

Due April 20 (Mon), 2015

## I. Image Filtering and Hybrid Images

- Read the following paper:
  - Oliva, Torralba, and Schyns. Hybrid images are static images that change in interpretation as a function of the viewing distance, SIGGRAPH 2006

The goal of this assignment is to implement an algorithm that can 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.

- **Image Filtering**:
  Implement your own code for image filtering, `my_imfilter()` in MATLAB or C++ (You will get extra credit if you use C++). It 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. Test your program on the provided images.
- **Hybrid Images**:
  A hybrid image is the sum (or average) of a low-pass filtered version of the one image and a high-pass filtered version of a second image. The cutoff-frequency of each filter should be chosen empirically. In the starter code, the cutoff frequency is controlled by changing the standard deviation of the Gausian filter used in constructing the hybrid images.
- For your favorite result, you should also illustrate the process through frequency analysis. Show the log magnitude of the Fourier transform of the two input images, the filtered images, and the hybrid image. In MATLAB, you can compute and display the 2D Fourier transform with: `imagesc(log(abs(fftshift(fft2(gray_image)))))`
- Try your algorithm on the sample image pairs provided. And try creating a variety of types of hybrid images of your own (change of expression, morph between different objects, change over time, etc.). The site has several examples that may inspire.
- Construct and show the image pyramid of your hybrid images using MATLAB function `imresize(image, 0.75, 'bicubic')` up to 5 levels.
- **Forbidden MATAB functions** which you cannot use in your final code: `imfilter()`, `filter2()`, `conv2()`, `nlfilter()`, `colfilt()`.

## II. Edge Detection

- Read the following paper on the bilateral filter.
  - C. Tomasi and R. Manduchi, "Bilaeral Filtering for Gray and Color Images", ICCV 1998
- Implement an edge detection algorithm that uses
  - the zero-crossings of the LOG (Laplacian of Gaussian)
    - Simple finding of all zero crossings may yield a lot of noisy edges. So, you need to threshold the edge strength (slope and magnitude) for detecting only strong edges
    - Do not round the filter output to the nearest integer value before computing zero crossings.
  - And the zero-crossings of the LOBG (Laplacian of Bilateral Gaussian)

    - You can derive the LOBG equation and design the kernel similarly to LOG case
- Write a single algorithm that has two kernel options, LOG and LOGB.
- Test your algorithm on the test images by varying the parameters of each algorithm (Try 9x9 kernel size with Gaussian std = 1.4).
- Add Gaussian noise with std= 20 to the test images and repeat the test on them.
- Show and compare your results, and discuss on them. Give any ideas for improving the performance.
- **Forbidden MATAB functions** which you cannot use in your final code: `edge()`.
- (Optional) Speed up by implementing the real-time bilateral filtering technique:
  Q. Yang, K-H, Tan and N. Ahuja, "Real-time O(1) Bilateral Filtering", CVPR2009

**Submission instructions: what to turn in**

- Upload the electronic file that includes the report, source code, and data in a single zip format with the name "ICV_assignment#1_yourname.zip" on the ETL class homepage.
- The report should include the brief description of the problems, results, and discussions

## References:

- Tutorial on MATLAB

**Academic integrity:**
Feel free to discuss the assignment with each other in general terms, and to search the Web for general guidance (not for complete solutions). Coding should be done individually. If you make substantial use of some code snippets or information from outside sources, be sure to acknowledge the sources in your report. At the first instance of cheating (copying from other students or unacknowledged sources on the Web), a grade of zero will be given for the assignment. At the second instance, you will automatically receive an F for the entire course.