HW3 linear filters (aka, the image as signal) CSx73 CV 19sp Johnstone due: 2.14.19

Question 1: hybrid images

Implement the hybrid images paper of Oliva, Torralba, and Schyns (see Readings). Given a pair of aligned images, a hybrid image is the sum of a low-pass-filtered version of one image and a high-pass-filtered version of the other image (p. 527-8). The high resolution image dominates at near distances, while the low resolution image dominates at far distances.

We have discussed low-pass filtering a lot in class, but high-pass filtering only a little bit (remember the sharpening mask?). This discussion of high-pass filtering may be useful. The key idea is that a Gaussian blur removes high frequencies from the image. This HW will help you explore how much.

I have provided you with 5 aligned image pairs in a tarball `image_pairs.tar`:
bicycle/motorcycle, albert/marilyn, cat/dog, fish/submarine, and bird/plane.
Extract with `tar xf img_hy.tar`.

As discussed in the paper, alignment is an important preprocessing step.
Bonus: create your own aligned image pair of your face with 2 expressions, and the associated hybrid image.

To explore the frequency landscape, you will display a grid (say 3x3) of the image using progressively larger sigma, both for low-pass filtered imagse and high-pass filtered images (see hw3.m)

Deliverable: the hybrid images (jpg) named by the pair (e.g., cat_dog.jpg); hw3_report.txt with sigma pairs for each image pair, your hw3.m solution (left set up for your favourite image pair).

Important note: the high frequency image (high-pass filtered) is zero mean and has negative values. Therefore, your images must be double over [0,1] to allow negative values. When visualizing the high frequency images (but only when visualizing), add .5 for better viewing.

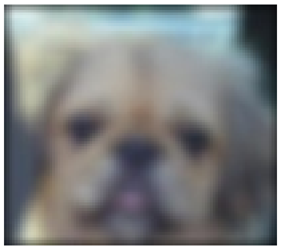
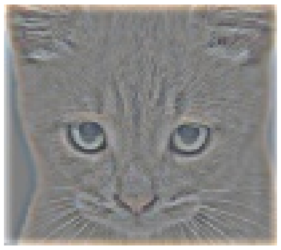
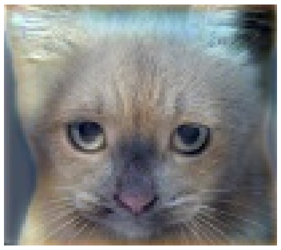
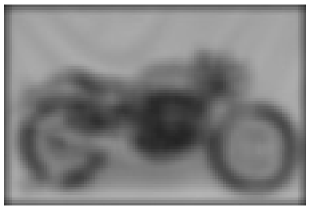
If you prefer, you may translate the filtered images to grayscale, yielding a grayscale hybrid image (see Examples).

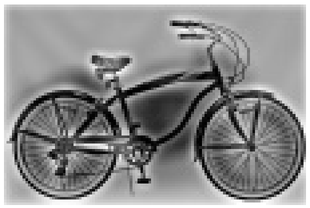
Discussion

Expressed in the frequency domain (what the paper calls the Fourier domain), the hybrid image sum is I1 . G1 + I2 . (1-G2), where I1 and I2 are the two images and G1 and G2 are two Gaussians, which are parameters that can be tuned for each image pair, controlling how much high frequency to remove from I1 and how much low frequency to remove from I2. How do you control this? By controlling the standard deviation σ of the Gaussian.

As context, the Fourier transform transforms a signal (say an image) from the time domain to the frequency domain. (You do not need to know anything about Fourier transform to implement hybrid images.) Convolution in the time domain is equivalent to multiplication in the frequency domain (a major incentive for the Fourier transform). And since the Fourier transform is linear, addition in the frequency domain is equivalent to addition in time domain. Therefore, the image sum in the time domain becomes I1 * G1 + I2 * (1-G2). You must choose G1 and G2 appropriately as part of your design. To allow you to control only one parameter σ, please set the hsize (width) of your Gaussian kernels to 31.

Examples



Reading

- **Oliva, Torralba and Schyns**. *Hybrid Images*, SIGGRAPH 2006, 527-532.
- **David Marr and Ellen Hildreth**. *Theory of Edge Detection*, Proceedings of the Royal Society of London, Series B, 1980, vol 207, 187-217.

673 only: prepare a 1-page report on the key ideas of the Marr-Hildreth paper

673 only: Complexity analysis of separable filters and convolution.

Note: Matlab is not allowed in this question. Please show all steps in order to get full credit. Throughout this question, $*$ stands for convolution. Convolution is defined as follows.

$$(I * F)[i, j] = \sum_{k,l} I[i-k, j-l]F[k, l]$$

.

a. Convolving the following I and F (specified in row major order). Assume we use zero-padding where necessary.
$$I = [2\ 0\ 1; 1\ \ -1\ 2] \quad F = [1\ \ -1; 1\ \ -1]$$

b. F is separable: $F = F_1 F_2$ where $F_1 = [1; 1]$ (a column matrix) and $F_2 = [1 - 1]$ (a row matrix). Computer the convolutions $(I * F_1)$, then $(I * F_1) * F_2$
(i.e., 1D convolution on each column, then 1D convolution on each row).

c. Prove that if a filter F is separable as $F = F_1 F_2$, then $I * F = (I * F_1) * F_2$. (Hint: expand the convolution formula.)

d. What is the big-O complexity of convolving an $n \times n$ image with an $m \times m$ kernel? What is the big-O complexity of convolving an $n \times n$ image with an $m \times m$ separable filter $F = F_1 F_2$ using $(I * F_1) * F_2$?