**EL-GY 6123 Image and Video Processing, Spring 2019**

**Programming Assignment 3: Pyramid and Wavelet**

This assignment is intended to review a) the concept related to image resizing, by asking you to generate a pyramid representation; b) multiresolution representation using pyramid; c) multiresolution representation using wavelet transforms; d) compare the representation efficiency of pyramid vs. wavelet; e) compare the denoising performance of softthresholding on Wavelet transform and using a linear filter.

1. Pyramid generation, reconstruction, and representation efficiency.
   1. Write a function that can simultaneously generate a 3-level Gaussian and Laplacian Pyramid for a given image. You could use the cv2.resize( ) function, both for down sampling, and for up sampling. Use the “INTER\_LINEAR” for downsampling filter; use the “INTER\_CUBIC” for upsampling filter. You can choose how to store and display the different levels of the pyramids.
   2. Write a function that can reconstruct an image from the Laplacian pyramid.
   3. Write a function that i) reads a gray scale image (or using only the luminance component of a color image), ii) generate the Gaussian and Laplacian pyramids using the function from part a); iii) Quantize the pixels in the Laplacian pyramid with a quantization stepsize *Q*. The dequantized value for a pixel in the Laplacian pyramid with value f can be determined from

*Q(f)=floor( (f-mean+Q/2)/Q) \*Q+mean*

You can assume mean = 0 for the Laplacian images and mean = 128 for the very top image of the pyramid.

iv) Count the number of non-zero pixels in the pyramid after quantization; v) reconstruct the image from the thresholded Laplacian pyramid using the function from part b); vi) Compute the PSNR between the reconstructed image and the original image using

*PSNR (dB)=10 log\_10 ( 255^2/MSE),* where MSE is the mean of the squared error between original and reconstructed pixels.

The function should take the image and threshold as input, and return the reconstructed image, the number of non-zeros and PSNR of the reconstructed image.

* 1. Write a main program that reads in an image, and calls the function in part c) repeatedly with different values for Q, with Q=2^n, n=0,1,…, and plot a curve of PSNR vs number of non-zeros. Such a curve evaluates the representation efficiency of the Laplacian pyramid. Here we use the number of non-zeros as a surrogate for the number of bits needed to describe the non-zero coefficients.

Note: for Parts c-d), if your functions in a-b) do not work properly, you can use the built-in functions in python to generate Gaussian and Laplacian pyramids and reconstruct images from the Laplacian pyramid.

1. Wavelet transform and reconstruction, and representation efficiency.
2. Write a function that can generate a 3-level wavelet transform of an image. You could use the function pywt.dwt2( ) to generate one-level wavelet decomposition. Choose Daubechies (db) wavelet and “symmetric” for boundary treatment. Ideally you would organize your data so that all the wavelet subimages together are stored in a 2D arrange as the original image.
3. Write a function that can reconstruct the original image from its wavelet transform subimages. You can use the function pywt.idwt2( ).

c-d) Repeat c-d) for Prob. 1 but on the wavelet subimages.

Note: for Parts c-d), if your functions in a-b) do not work properly, you can use the built-in functions in python for wavelet transform and inverse transform.

1. Wavelet Denoising

Write a main program that i) reads in an image, ii) add Gaussian random noise with a noise standard deviation at 5% of the image dynamic range; iii) Generate the wavelet image using your function from Prob. 2(a); iv) Perform denoising using soft thresholding on the Wavelet image. For any coefficient with value f, and a threshold T, the soft thresholded value is defined as

Recall that the threshold T should be chosen based on the noise variance and the wavelet coefficient variance, following where is the noise variance, is the variance of the wavelet coefficients. Note that the wavelet coefficient variance differs among the subbands. Your program should estimate the coefficient variance for each band and use the corresponding threshold when performing soft thresholding on the wavelet image. v) Reconstruct the denoised image from the thresholded Wavelet image using your function from Prob. 2(b). vi) Evaluate the quality of denoising by computing the PSNR of the original image (without noise) and denoised image. Note that such quantitative evaluation is not feasible in practice as you will not have access to the original image.

Note: If your functions in Part 2(a-b) do not work properly, you can use the built-in functions in python for wavelet transform and inverse transform.

1. For the same noise-added image, try to reduce the noise using the following smoothing filter and compute the PSNR between the denoised image and the original image.

Discussion:

1. Based on the results in Prob. 1(c) and 2(c), compare the representation efficiency of Laplacian pyramid vs. wavelet transform.
2. Based on the results in Prob. 3 and 4, compare the denoising performance of the two methods. Pay attention not only to noise suppression but also detail preserving.

Optional: You can try different down-sampling and interpolation filters for pyramid representation; and different wavelet transforms, to see their effects on representation efficiency and denoising. You can also try different noise levels.

Optional: Investigate the representation efficiency of block DCT (e.g. DCT over every 8x8 blocks) by following similar framework, and compare it to Wavelet and Laplacian pyramid.