E9 241: DIGITAL IMAGE PROCESSING

ASSIGNMENT 7

Topics: Image Pyramid, Wavelet/Filterbank Transforms, Interpolation, Principal Component Analysis, Radon Transform

(1) Write Matlab/Python scripts to implement the Gaussian and Laplacian pyramid decomposition of a given image. The program should accept the number of levels of decomposition as the input parameter and display side-by-side the images at each level of the Gaussian and Laplacian pyramids. The standard deviation of the Gaussian should be adjustable using a slider and the results should be updated automatically. Report your results on the *Van Gogh* image. You could use the iterated moving-average smoother from the previous assignments for an efficient implementation of the Gaussian smoother.

(2) Use the Laplacian pyramid to fuse/blend the *Apple* and *Orange* images. Follow the procedure outlined in `http://persci.mit.edu/pub_pdfs/RCA84.pdf`.

(3) Write a Matlab/Python script to implement an n-level Haar decomposition and synthesis. The parameter n takes values 1, 2, 3, or 4. Incorporate a radio button to select the number of levels. Show the analysis and synthesis results at each level of the representation. Show the difference image as well (difference between the input image and the synthesized image). Compute the maximum, minimum, and the standard deviation of the difference image. Report your results on the *Peppers* image.

(4) Write a Matlab/Python script to implement bilinear interpolation and cubic B-spline interpolation to upsample an image by a factor of two along x and y directions. Report your results on the *Cameraman* image

(5) Write a Matlab/Python script to rotate an image by an arbitrary angle (user-specified parameter). Zero-pad the given image to begin with so that the rotated image always fits within the dimensions of the zero-padded image regardless of the angle of rotation. Your program must have the option of selecting bilinear or cubic B-spline interpolation for implementing the rotation.

(6) Consider the *Yale Face database* and assume that each face image is a vector in a high-dimensional space. Write a Matlab/Python script to compute the covariance matrix of the entire dataset and display the top 20 principal components as images (*eigenfaces*). Also, show the distribution of the eigenvalues of the covariance matrix. Extend the program to progressively reconstruct an approximation of the ground truth image using the first n principal components (for $n = 1, 2, 3, \ldots$). Display the original image, the reconstructed image, and the error image. Label the images appropriately.

(7) Generate a 256×256 image with three impulses at arbitrary, but known, locations. Write a program to generate the corresponding *sinogram*. Consider 20 uniformly spaced angles. Next, write a program to perform reconstruction based on filtered backprojection using the *Ram-Lak* filter. Display the image, the sinogram, and the reconstructed image.
