CSE 509

Homework 2

Multi-resolution Image Decompositions

Submit MATLAB/other source code, which when run displays the required outputs and your analysis and comments in a report properly formatted in MS Word and submitted in PDF file. You can choose the input images from the ones provided in folder HW2_images.

1 - *Gaussian Pyramid (Filtering, Decimation) [40pt]:*

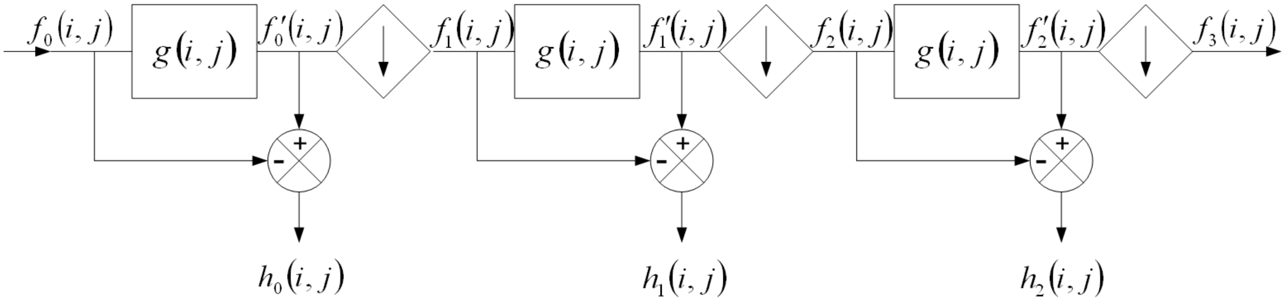
Using image **calendar.jpg** in folder **multires**, do the following:

a) Crop the image so that the length and width are powers of 2, extract the gray level image for the following steps.

b) Define a *5x5* circularly symmetric Gaussian filter and use it to construct a *3-level* Gaussian pyramid. To construct the pyramid, you need to filter the given image by the Gaussian filter. Then, using function *imresize*(), apply *2x2* subsampling (i.e. downscale by 0.5) in both horizontal and vertical directions to form the second level of the pyramid. Repeat the process to form the third level.

c) Do the same by using a *7x7* box-car filter, i.e. (1/49) * *ones* (7,7). Look at the outputs of each level and explain which one is better and why?

2 - *Laplacian Pyramid (Image Differencing)[30pt]:*



In order to construct a Laplacian pyramid, we simply take the difference between the Gaussian filtered image and the unfiltered image in the same level of Gaussian pyramid to form difference images *h_0*, *h_1*, *h_2* at each level as shown in the Figure above. Thus, a *4-level* Laplacian pyramid consists of one low resolution picture and three successively larger difference images, i.e., $\{h_0, h_1, h_2, f_3\}$.

Recover the full-resolution image by adding the difference images to the interpolated images at each level successively. Include the two images, original and reconstructed image in your report.

3 – *Bilateral filter tune-up [30pt]:*

Using the image **couple.jpg** in folder bilateral and filter using Matlab's *imbilatfilt*() carry out the following:

- Read the image and convert it to double precision gray level with values normalized to 0.0 to 1.0
- Filter the gray level image using the default values for arguments *degreeOfSmoothing* and *spatialSigma*. Read the Matlab help for *imbilatfilt* to learn the details, and to learn how to change those parameters. Use *imshow()* to display the filtered image, and inspect the results.
- Recover the filtered RGB image:
    - Convert the original RGB image to Ycbcr (use *rgb2ycbcr*(), and replace the Y component with the filtered gray level image. Be careful to convert the filtered gray level back to uint8 with max value 255, before replacing.
    - Display the filtered RGB image and describe the results
- Improve the filtered RGB image: since you figured out the default parameters, now go ahead and experiment making changes to them (increase the param values up/down one at a time and together) to modify the result observed in the gray level using default params. When you get a result that you think is better (e.g. more noise reduction and/or better edges), do as before to put the improved gray level image back into the original RGB and see the improvement in RGB. Compare the original RGB, the default bilateral filtered RGB, and the improved RGB using your parameters.

What to submit

1. Matlab/Python code: Your code must run to display the required output images
2. Your PDF report must include snapshots of the images and your answers, observations, and comments as required for each question

HINTS:

- Review the theory of multiscale representations and bilateral filter before doing the work. See also https://medium.com/jun94-devpblog/cv-4-multi-scale-representation-gaussian-and-laplacian-pyramid-527ca4c4831c for further information.
- When done, try out with the other images provided, so you can improve your knowledge of the subject.