E9 241 Digital Image Processing

Assignment 04

Due Date: November 10, 2022 - 11:59 pm **Total Marks:** 90

Instructions:

For all the questions, write your own functions. Use library functions for comparison only.

- Your function should take the specified parameters as inputs and output the specified results.
- Also provide the wrapper/demo code to run your functions. Your code should be self-contained, i.e., one should be able to run your code as is without any modifications.
- For python, if you use any libraries other than `numpy`, `scipy`, `scikit-image`, `OpenCV`, `pillow`, `matplotlib`, `pandas` and default modules, please specify the library that needs to be installed to run your code.
- Along with your code, also submit a PDF with all the results and inferences. Include answers to subjective questions, if any.
- Put all your files into a single zip file and submit the zip file. Name the zip file with your name.
- **Vectorize your code. Non-optimized code may be penalized.**

---

1. **Image Downsampling:**

   (a) Downsample the image `city.png` by a factor of 2, 4 and 5. For the factor of 2, downsample the image by selecting every second pixel in both directions (do not use library functions to downsample). Apply a similar procedure for the other factors. What do you observe and Why?

   (b) Now, first filter the image with a spatial domain Gaussian Low Pass Filter before downsampling the image. You can use a 5×5 window and $\sigma = 2$. What difference do you notice, and why? Compare your result with a library function.

   For the factor of 5, find out the optimal window size and σ value that minimize the mean squared error between your output and the library function output.

   (10+15 Marks)

2. **Edge Detection:** For the grayscale images Checkerboard.png, NoisyCheckerboard.png, Coins.png and NoisyCoins.png,

   (a) Smooth the input image using a spatial domain Gaussian filter. You can use a 5×5 window and $\sigma = 5$.

   (b) Check the results of the edge detection with and without Gaussian smoothing. Use sobel/prewitt operator to calculate gradients and apply threshold on gradient magnitude to get the edges.

   (c) Are the edges detected if the images are rotated?

   (d) For the same set of images, use Laplacian operator to detect the edges. For the noisy images, threshold the filtered image before finding the zero crossings.

   Analyse the difference between first-order and second-order gradient-based edge detectors for clean and noisy images.

   (5+10+5+10 Marks)

3. **Interest Point Detection:** For the images Checkerboard.png and MainBuilding.png,

   (a) Implement Harris corner detector. Experiment with different threshold values and report your results.

   (b) Modify the images by rotating, scaling and adding noise. Now detect corners in the modified images. Analyse the difference in performance as compared to the original images.

   (c) In addition to the real images, create a synthetic image with an edge, a blob, and a corner (both straight and rotated). See which of the above are detected using Harris corner detector by changing the threshold.

(15+5+15 Marks)