**CMP SCI 6420** **Image Processing and Computer Vision**
Assignment # 3 Due Date: March 16, 2023

---

# Image Restoration

### Purpose

The purpose of this assignment is to implement the adaptive filters and determine their effect in restoring an image.

### Task

Write a function to add salt-and-pepper noise into an image. You should be able to control the amount of noise added, by specifying the percentage of pixels to get affected by noise. Make sure that you save a copy of the original image $f(x, y)$ as well.

Write the adaptive mean filter and the adaptive median filter (from first principles) and apply those filters on your corrupted image, yielding the restored image $\hat{f}(x, y)$. Find the Euclidean distance between $f(x, y)$ and $\hat{f}(x, y)$ and print it, alongside the restored image, for both the adaptive filters.

You may work with just the grayscale images.

### Invoking the solution

Your solution will be invoked using the following command:

```
adaptive [-h] [-n N] image_file [corrupted_image] [output_mean_file] \
[output_median_file]
```

| | |
|---|---|
| `adaptive` | Name of your executable |
| `N` | Percentage of corrupted pixels; Range [1,90]; Default 10 |

The parameters enclosed in `[]` are optional. If the output files are not specified, develop a convention that will convey the meaning from the input `image_file`. The output mean and median files indicate the restored versions using the corresponding adaptive filter.

### Suggested implementation steps

1. Parse the command line. You can create your own parser or use the class `CommandLineParser` provided by OpenCV. If the user specifies the option `-h`, print a help message and exit. Otherwise, assign the suggested parameters from user inputs or default values.
2. Read the input image. Copy it into another image and add noise.
   (a) Create two images of the same size and number of channels as the input image. One of them will be the salt image and the other will be the pepper image. In salt image, all pixels will have the value `0X00` while in pepper image, all pixels will have the value `0XFF`.
   (b) Use a random number generator to randomly change the specified percentage of pixels to `0XFF` in salt image and to `0X00` in pepper image.
   (c) Use bitwise and to add pepper noise and bitwise or to add salt noise.
3. Write the function to perform adaptive mean filtering.

4. Write the function to perform adaptive median filtering.
5. Apply the two filters separately on the noisy image and save the restored images.
6. Find the Euclidean distance between the restored images and the original saved image.
7. Display all images and print the Euclidean distances computed.

**Criteria for Success**

You should be able to clean up the corrupted image to a certain extent. You can see the amount cleaned by comparing the Euclidean distance of the restored image to that of the corrupted image from the original image.

**Grading**

I'll use the following rubric to assess your submission.

1. *Overall submission; 30pts* Program compiles and upon reading, seems to be able to solve the assigned problem.
2. *Command line parsing; 10pts* Program is able to parse the command line appropriately, assigning defaults as needed; issues help if needed.
3. *Addition of salt and peppernoise; 10pts* Program is able to add addpropriate amount of salt-and-pepper noise.
4. *Adaptive mean filtering; 20pts* Program restores the image, at least subjectively, to a certain extent.
5. *Adaptive median filtering; 20pts* Program restores the image, at least subjectively, to a certain extent.
6. *Euclidean distance computation; 10 pts* Euclidean distance is computed correctly.

**Submission**

Submit an electronic copy of all the sources, `README`, Makefile(s), and results. Create your programs in a directory called *username*.3 where *username* is your login name on delmar. This directory should be located in your `$HOME`. Once you are done with everything, *remove the executables and object files*, and issue the following commands:

```
% cd
% chmod 755 ~
% ~bhatias/bin/handin cs6420 3
% chmod 700 ~
```

Do not copy-paste these commands from the PDF; type in those commands.