CMP SCI 6420 **Image Processing and Computer Vision**

Assignment # 3 Due Date: March 09, 2021

Frequency Domain Filtering

Purpose

The purpose of this assignment is to remove periodic noise in an image by filtering in the frequency domain. You will get to use the OpenCV functions to convert an image into frequency domain by applying Discrete Fourier Transform (DFT), apply the filter, and convert the filtered image back into spatial domain by applying inverse DFT.

Task

The periodic noise is best removed by filtering an image in the frequency domain. The periodic noise may have been added by atmospheric disturbance or sensor aberrations. Your task is to remove the periodic noise in a given image.

Assume that you are given an image with periodic noise. Read in the image and convert it to frequency domain. Display the converted image. Using the mouse, specify the frequencies that need to be remove and change them to zero. Apply inverse transform to convert the image back in spatial domain and display it.

Invoking the solution

Your solution will be invoked using the following command:

```
freq_filter [-h] input_image [output_image]
```

`freq_filter`	Name of your executable
`input_image`	Name of file containing image with periodic noise
`output_image`	Name of file to save the image; select some default name if this name is not specified.

The parameters enclosed in `[]` are optional. You are free to use long parameter names such as `--help` for `-h`.

Suggested implementation steps

1. Parse the command line. You can create your own parser or use the class `CommandLineParser` provided by OpenCV. Each of the optional arguments may have a default value. 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 as grayscale.
3. Compute optimal size for DFT by using the function `cv::getOptimalDFTSize`. Convert the input image to optimal size by using the function `cv::copyMakeBorder` assigning the value 0 to added pixels. Call the new image as padded image.
4. If necessary, add 1 to number of rows/columns in padded image to make them even. This is needed to divide the image neatly into four quadrants later.
5. Create two floating point images (`CV_32F`) of the same size as padded image. These images will form the real and imaginary planes in a complex image. Copy the padded image into the real plane, making the imaginary plane all zeroes.

6. Apply DFT to the complex image using function `cv::dft`.
7. Consider the DFT image to be made up of quads. Swap the quads along the diagonals as shown in the following figure.

q0	q1	⇒	q3	q2
q2	q3		q1	q0

   This swapping ensures that the center of the image is now the origin.
8. Use `cv::split` to split the complex image output by DFT into real and imaginary planes. Use the two planes to compute the magnitude of the image, using `cv::magnitude`. Make sure that the magnitude image is different from the complex image output by DFT.
9. Normalize the magnitude image to $[0, 1)$. Multiply it by 255 and convert to 8-bit image (`CV_8U`).
10. Display the magnitude image to visualize the frequencies. Find the location of periodic frequencies and change those to zero. Threshold the image such that all the non-zero pixels are changed to 1.
11. Multiply the two planes of complex image with the thresholded magnitude image. That will remove the frequencies corresponding to noise.
12. Swap the quadrants again using the above figure.
13. Apply inverse DFT to the complex image using `cv::idft`.
14. Split the planes in the complex image, normalize the real plane, multiply it by 255, and convert it to 8 byte integer (`CV_8U`). Display/save the resulting image.

Criteria for Success

You should be able to successfully perform the steps outlined above and remove noise from the given 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. *Apply DFT/iDFT; 50pts* Program is able to apply the transforms appropriately and successfully.
4. *Removing periodic noise; 10pts* Successfully find and change to zero the frequencies corresponding to noise.

It will be nice if you can automatically discover and remove the frequencies corresponding to noise. I observed that the magnitude image consisted of a number of 0s and frequencies were clustered. You can perform cluster analysis by using the functions such as `cv::findContours` on the magnitude image and changing the noise contours to zero. While working with `findContours`, please make a deep copy of the image because it may corrupt the image.

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.