Programming Assignment 3
Fourier Transform Implementation

Note: Use simple images like (apple.ppm, beetle.ppm, rice.ppm) for testing your application since computing Fourier transform using Direct method (Brute-force method) may take a lot of time on your computer.

Objective:
In this assignment, you will add functionality to the GUI from Assignment 2. You will implement the Fourier transform (forward and backward) using the direct method and the Fast-Fourier Transform (decimation-in-time method).

Part 1 - Direct Implementation (10 points)
In this part of the assignment, implement the code for calculating 2D Fourier Transform using the following methods:

(a) Direct method - two-nested for loops (4 points)
Implement the DFT as per the definition of the formula. You will implement two for loops that march along the rows and columns of the input image FOR EVERY POINT IN THE OUTPUT. That means, you will march through the entire image for every pair of values (u,v) in the output range (which is considerably worse than the part (b) below.)

(b) Separable method - As a concatenation of two 1D Fourier Transforms. (6 points)
You will first perform the 1D Fourier Transform on the rows of the image, store the intermediate result, and then perform the column wise Fourier transform on the intermediate values to complete the calculation of the 2D transform.

Part 2 - Fast Fourier Transform (20 points)
In this part of the assignment, you will implement the time-decimation Fast Fourier Transform as discussed in class. Some helpful hints to develop the functionality:

(a) You will need to pre-process the input data in order to align the data points according to the Butterfly configuration. Use the bit-reversal algorithm to do so.
(b) Your output should be an in-place computation (to save memory).
(c) You will need to post-process the output of the FFT (Complex Numbers) into a scalar representation in order to display it as an image. You may use the "log" transform to achieve this and then display the Fourier Spectrum.

For both the parts (1) and (2), you should display the following:
(a) The original image - displayed normally in the application window.
(b) The Fourier Spectrum - displayed in a separate window/panel.
(c) The Reconstructed image - this image is calculated by applying the inverse Fourier transform to the Fourier spectrum. Ideally, this should be the same image as the input image. Display this image under a separate window/panel as well.

Some implementation points to consider:
- You will save considerable coding effort if you could reuse your forward Fourier Transform code to calculate the inverse transfrom as well. **How will you implement the Inverse Fourier transform using the Forward Fourier Transform?**
- Your Fourier Spectrum should be centered in the location (M/2, N/2) and not at (0,0). **How will you shift the Fourier Spectrum F(u,v) by manipulating the input signal f(x,y)?**

Part 3 - Extra Credit (5 points)
(a) Use intensity / color slicing instead of a "log" transform to display the Fourier Spectrum in pseudocolor representation. In this case, you will represent a range of values in the Fourier spectrum by a single color - say [0, 10.0) is red, [10.0 to 100.0) is dark blue etc. The user should be able to input the number of "bins" for slicing. Provide examples in the README document. (2.5 points)

(b) Implement a trivial compression scheme by allowing the user to eliminate some frequency components. For e.g. Implement a thresholding scheme that only allows certain values of (u,v) to pass through. The user may manipulate two sliders (in the GUI) to choose the lower and upper range of the frequencies that are passed through. You should zero-out the frequency values outside this range when computing the inverse Fourier transform. How many frequency components can you eliminate before you start losing the image details and the output is no longer similar to the original? Provide image examples in the README document. (2.5 points)

Deliverable: For submitting your assignment, create an archive that is named as follows:

CS370_*<student_login_name>_<assignment#>***.zip**

Example: If your login is pacquiao.mandy and assignment 2 is being submitted, your folder would be named CS370_pacquiao.mandy_2.

Include the following in the archive:

a) A folder named **src** – It contains complete source code of the application including any project/solution files that you create. The instructor will not be responsible for creation of Makefiles or VS Projects/Solutions to make your project work. Instructor will compile your code and if your code does not compile to produce an executable, it will not be graded.

b) A folder named **doc** - It contains README.docx having a short description of your solution to the programming project. This includes description of extra credit if you have attempted it. In addition, compare and contrast the runtimes of all three algorithms on **at least 5 images from the dataset**. You should submit the document as PDF file. If this document is not in proper format, or the sample output images are not from your implementations, then you will get ZERO grades for this assignment. Use excel charts to visualize the timing disparities. If you have implemented the extra credit part (b), comment on the distribution of "power" in the image as we move from the origin (u=0 and v=0) towards higher frequency components on the edge of the image.

c) A folder named **release** – It contains your executable. The instructor should be able to click and run executable to test your application.

Your project should compile and execute without errors or warnings in the latest Visual Studio environment on Digipen computers.

Failure to submit the deliverable in the required format, or not provide enough examples in the README file, will result in a zero assignment grade.

Late Submission: Refer to the course outline for the late submission guidelines.

Grade Sheet for Assignment-3:

Item	Points Possible	Points Awarded	Comments
Part 1			
Direct method	4		
Separable method	6		
Part 2			
FFT Implementation	20		
Part 3			
Extra Credit – Intensity slicing	2.5		
Extra Credit – Image compression	2.5		
Penalties, if any			
TOTAL GRADE	**30**		

Important Note:

Always rebuild the solution whenever there is modification in the code. The instructor should be able to rebuild/execute your project on any computer at DigiPen having the following software installed:

1. **Visual Studio Community 2022**
2. **wxWidgets-3.0.5**