University of Engineering & Technology, Mardan

Department of Electrical Engineering

Assignment #02

Subject: EE411 Digital Image Processing — Instructor: Dr. Irfan Khan

Due Date: 07/05/2025 — Total Marks: 20 [CLO 1: PLO 1, C2]

NOTE: Copied quizzes will render zero marks.

Take a black & White image of yourself. Write a MATLAB code to perform:

a. Load the image in MATLAB
b. Multiply the image by $(-1)^{x+y}$
c. Computing the DFT using the equation shown below. Do not use the built-in Matlab function (FFT2)

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

d. Take the complex conjugate of the transform $F(u,v)$
e. Compute the inverse DFT
f. Multiply the real part of the result by $(-1)^{x+y}$
g. Display the original and the final images side by side.

Matlab code:

```
clear all
close all
clc
%% Step a: Load the image
img = imread('My pic.jpg');  % Replace with your image name
if size(img,3) == 3
   img = rgb2gray(img); % Convert to grayscale if RGB
end

%% Resize for faster testing (optional)
maxSize = 256;
if max(size(img)) > maxSize
   img = imresize(img, [maxSize maxSize]);
end

img = double(img);
[M, N] = size(img);

%% Step b: Multiply image by (-1)^(x+y)
[x, y] = meshgrid(0:N-1, 0:M-1);
```

```
img_shifted = img .* (-1).^(x + y);

%% Step c: Compute the DFT using matrix multiplication (no fft2)
u = 0:M-1;
v = 0:N-1;

fprintf("Computing DFT and inverse DFT. Please wait...\n");
tic; % Start timer

Wm = exp(-2i * pi * (u' * u) / M); % M x M DFT matrix
Wn = exp(-2i * pi * (v' * v) / N); % N x N DFT matrix
F = Wm * img_shifted * Wn;

%% Step d: Take complex conjugate
F_conj = conj(F);

% Step e: Compute the inverse DFT
Wm_inv = exp(2i * pi * (u' * u) / M);
Wn_inv = exp(2i * pi * (v' * v) / N);
img_reconstructed = (Wm_inv * F_conj * Wn_inv) / (M * N);

toc; % End timer

%% Step f: Multiply real part by (-1)^(x+y)
final_img = real(img_reconstructed) .* (-1).^(x + y);

%% Step g: Display original and final images with adjusted contrast
figure('Name','Image Comparison','NumberTitle','off');
subplot(1,2,1); imshow(uint8(img)); title('Original Image');
subplot(1,2,2); imshow(uint8(mat2gray(final_img) * 255)); title('Processed Image');

% Optional: Save final result
% imwrite(uint8(mat2gray(final_img) * 255), 'processed_image.png');
```

Result: