ECE 429/529 Digital Image Processing

Fall 2020

Dept. of Electrical and Computer Engineering **Miami University**

Homework 8 Assignment

Weights: 100 pts
Due Date: November 11, 2020.
Turn in Method: Upload your homework by the midnight of November 11.
Note: Please turn in **every program** and **single document** including every figure (with an appropriate caption) and descriptions about how you process distorted/corrupted images.

1. Download 'HW8.jpg' from the class site. This image is distorted in the spectrum domain as described by Equation (5.77) in the 4th-edtion textbook (also shown in slide #40 of lecture note #13). The DC component is at the center of spectrum. The value I used are: T=1, 0<a<0.2, and 0<b<0.2. This image is also corrupted by very small Gaussian noise and quantization noise. **Note:** when creating distortion function pay attention to the situation that (ua+vb) equals to zero.

   Follow instructions in lecture notes to use (a) full inverse filter (b) partial inverse filter (c) Wiener filter, (d) Constrained Least Squares filter, and (e) Geometric Mean filter to compensate the image. You are NOT allow to use Matlab's advanced filtering functions for this homework. You will need to guess at values of a, b, and signal-to-noise ratio (the noise power is very small). Since I did not zero pad image when I distorted this image, do not zero-pad your image when conducting equalization. These five filters must be different filters. For example, you are not allowed to use the same filter for part (c) and (e). **Note:** When processing this image in frequency domain, do not zero-pad image and DC should be at the center of spectrum. You can use fftshift in your program.

   The code used to generate distorted image (you need to guess the values of *a* and *b*, and signal-to-noise ratio (related to *k*).

```
Orig_Spec=fftshift(fft2(B));
[N,M]=size(B);
S=size(B);
center=[N,M]/2;
for k1=1:N
    for k2=1:M
        arg=pi*((k1-center(1)-1)*a+(k2-center(2)-1)*b)+eps;
        H(k1,k2)=(T/(arg))*sin(arg)*exp(-1j*arg);
    end
end
Dis_Spec=H.*Orig_Spec;
dis_imag=ifft2(fftshift(Dis_Spec));
orig_dis_imag=dis_imag;
dis_imag=dis_imag+randn(size(dis_imag))*k;
dis_imag=uint8(dis_imag);
imwrite(dis_imag,'HW8.jpg','jpg');
```