XỬ LÝ ẢNH - 03/2025

**Tên sinh viên (số thứ tự trong danh sách), ví dụ như sau:**

1. Trần Anh Toàn (31)

2. Trịnh Minh Việt (35)

3. Trần Lê Long Vũ (36)

**Phần A (***Trả lời các câu hỏi sau đây mà không cần giải thích)*

1. Ta thực hiện phép biến đổi Y = f(I), với I là ảnh bên trái và Y là ảnh bên phải, hàm f tác động lên từng pixel của ảnh xám I (đầu ra tại mỗi pixel chỉ phụ thuộc vào giá trị của chính pixel đó). Giá trị mức xám của I thuộc [0,1]. Khi hiển thị, ảnh I và Y được hiển thị trong khoảng [min, max] của từng ảnh (ví dụ, dùng lệnh *imshow(I,[])*). Hàm f có dạng nào trong số các đáp án A, B, C, D?

|  |  |  |  |
| --- | --- | --- | --- |
| I  C:\___DATA_R20231203\EDU\Course\DTVT_XLA\Exam\Tracnghiem\7\in.png | | Y  C:\___DATA_R20231203\EDU\Course\DTVT_XLA\Exam\Tracnghiem\7\out.png | |
| A) | B) | C) | D) |

Trả lời: C

2. Cho 3 ảnh xám I1, I2, I3, và 3 histogram H1, H2, H3 của 3 ảnh này, nhưng không tương ứng (ví dụ, H1 chưa chắc là histogram của I1). Xếp lại theo từng cặp ảnh-histogram tương ứng.

|  |
| --- |
| C:\___DATA_R20231203\EDU\Course\DTVT_XLA\Exam\Tracnghiem\9\hist.png |

Trả lời: I1-H2, I2-H3, I3-H1

3. Xét ảnh xám I, bộ lọc F và G như dưới đây. Lọc ảnh I bằng bộ lọc F, cho ra ảnh X. Lọc ảnh I bằng bộ lọc G, cho ra ảnh Y. Histogram H1, H2, H3 của 3 ảnh I,X,Y được cho bên dưới (nhưng không lần lượt tương ứng, ví dụ H1 chưa chắc là histogram của I). Xếp thành từng cặp ảnh-histogram tương ứng.

|  |
| --- |
| K = 7; F = ones(K,K)/(K\*K);  K = 21; G = ones(K,K)/(K\*K);  C:\___DATA_R20231203\EDU\Course\DTVT_XLA\Exam\Tracnghiem\17\hist.png |

Trả lời: I-H2, X-H1, Y-H3

Khi mà kernel càng rộng thì bên phải hẹp dô, bên trái rộng ra

4. Xét ảnh xám 3 bit, kí hiệu là I, kích thước 4 như dưới đây. Cân bằng histogram đối với ảnh I. Gọi Y là ảnh thu được sau khi cân bằng histogram đối với I. Tìm giá trị Y(3,2), (với chỉ số được đánh theo Matlab, tức tính từ 1).

Trả lời: 4

**Phần B** (*Phần trả lời của các câu hỏi sau cần được giải thích ngắn gọn, minh họa, chèn code như bên dưới*)

1. Ta sẽ xem xét phép lọc trong miền tần số và xét tần số chuẩn hóa trong khoảng [-0.5, 0.5).

Lọc ảnh xám I bằng bộ lọc thông thấp lí tưởng với tần số cắt D. Thay đổi D và quan sát kết quả của phép lọc. Mô tả ngắn gọn kết quả thu được, minh họa bằng hình ảnh và giải thích. (Chú ý hiển thị cả ảnh I và phổ biên độ của ảnh I để dễ đánh giá).

*[Code]*

I = imread('cameraman.tif');

gray\_img = im2double(I);

F = fft2(double(gray\_img));

F\_shifted = fftshift(F);

[M, N] = size(gray\_img);

H\_ideal = zeros(M, N);

D\_0 = 100;

for i = 1:M

for j = 1:N

D = sqrt((i - M/2)^2 + (j - N/2)^2);

if D < D\_0

H\_ideal(i, j) = 1;

end

end

end

SY = F\_shifted .\* H\_ideal;

filtered\_img = real(ifft2(ifftshift(SY)));

figure;

subplot(2, 3, 1);

imshow(gray\_img, []);

title('Ảnh gốc');

subplot(2, 3, 2);

imshow(mat2gray(log(1 + abs(F\_shifted))));

title('Phổ ảnh gốc');

subplot(2, 3, 3);

imshow(H\_ideal, []);

title('Bộ lọc thông thấp');

subplot(2, 3, 4);

imshow(mat2gray(log(1 + abs(SY))));

title('Phổ sau khi lọc');

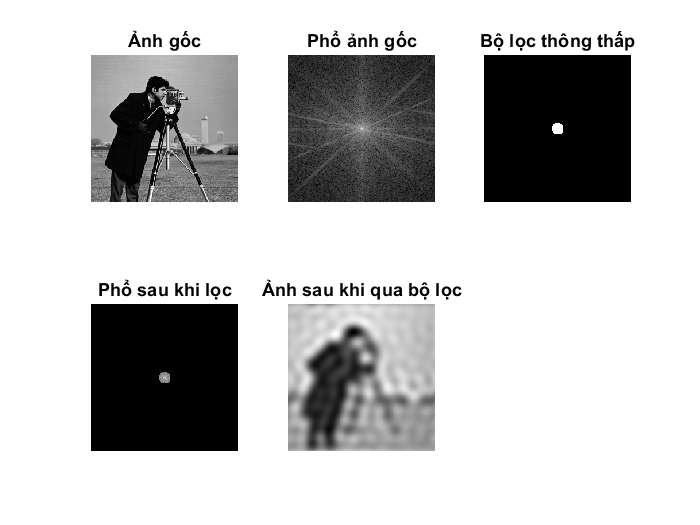
subplot(2, 3, 5);

imshow(filtered\_img, []);

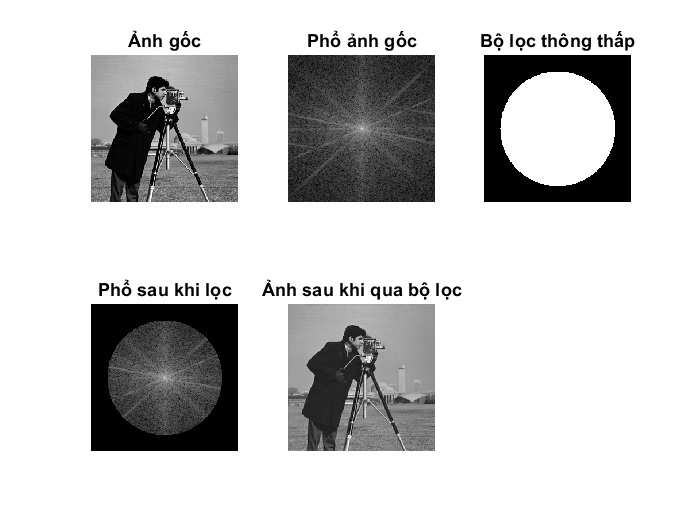
title('Ảnh sau khi qua bộ lọc');

*[Hình/bảng]*

D=10:



D=100:



*[Giải thích]*

- Nhận xét:

- Khi D = 10, ảnh bị làm mờ nhiều và xuất hiện các vân sáng tối xe kẻ

- Khi D = 100, ảnh giữ lại nhiều chi tiết hơn nhưng vẫn bị làm mờ nhẹ so với ảnh gốc

Vậy ta có thể thấy :

* Giá trị D càng nhỏ thì ảnh càng mờ, mất nhiều chi tiết vì bộ lọc thông thấp chỉ giữ lại các tần số thấp hơn tần số cắt. Khi biến đổi IFFT lại sẽ bị mờ hơn
* Giá trị D càng lớn thì ảnh càng rõ, càng có nhiều chi tiết vì bộ lọc cho phép các tần số cao hơn đi qua. Khi biến đổi IFFT lại sẽ rõ hơn
* Khi giảm D đến một giá trị nào đó thì xuất hiện các vân sáng tối xen kẻ đặc biệt là ở biên. Vì bộ lọc này là hàm sinc ở hàm thời gian và lấy chập với các điểm ảnh nên sẽ có gợn, và ở bên có sự thay đổi tần số lớn hơn nên dễ dàng thấy được các vân sáng tối xen kẻ so với uniform

Nếu thay bộ lọc thông thấp lí tưởng bằng bộ lọc thông thấp Butterworth và bộ lọc thông thấp Gaussian, và thực hiện lại quá trình lọc ở trên, thì có gì khác biệt trong kết quả thu được? Minh họa kết quả một cách hợp lý và giải thích.

*[Code]*

clc; clear; close all;

I = imread('cameraman.tif');

I = im2double(I);

[M, N] = size(I);

D0 = 15;

H\_ideal = zeros(M, N);

H\_butterworth = zeros(M, N);

H\_gaussian = zeros(M, N);

n = 2;

for u = 1:M

for v = 1:N

D = sqrt((u - M/2)^2 + (v - N/2)^2);

if D <= D0

H\_ideal(u, v) = 1;

end

H\_butterworth(u, v) = 1 / (1 + (D / D0)^(2 \* n));

H\_gaussian(u, v) = exp(-(D^2) / (2 \* D0^2));

end

end

I\_fft = fftshift(fft2(I));

I\_ideal = real(ifft2(ifftshift(I\_fft .\* H\_ideal)));

I\_butterworth = real(ifft2(ifftshift(I\_fft .\* H\_butterworth)));

I\_gaussian = real(ifft2(ifftshift(I\_fft .\* H\_gaussian)));

figure;

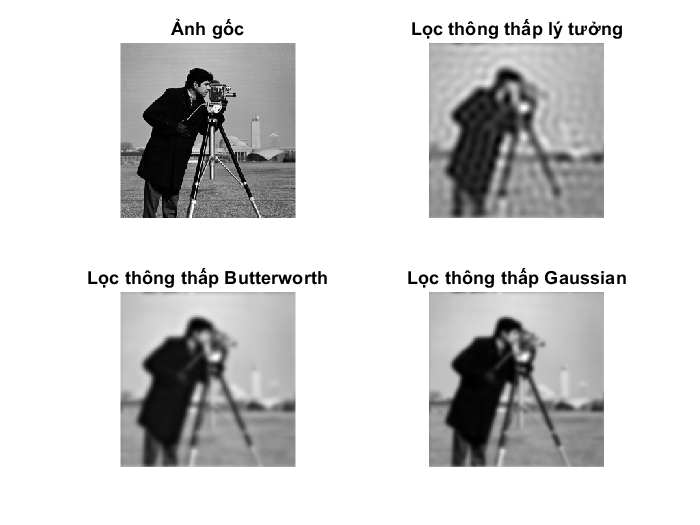
subplot(2,2,1); imshow(I, []); title('Ảnh gốc');

subplot(2,2,2); imshow(I\_ideal, []); title('Lọc thông thấp lý tưởng');

subplot(2,2,3); imshow(I\_butterworth, []); title('Lọc thông thấp Butterworth');

subplot(2,2,4); imshow(I\_gaussian, []); title('Lọc thông thấp Gaussian');

*[Hình/bảng]*



*[Giải thích]*

* Nhận xét:

 **Lọc thông thấp lý tưởng:** Ảnh có hiện tượng vân sáng tối xen kẻ gây ra do cắt tần số đột ngột.

 **Lọc thông thấp Butterworth:** Ảnh mờ dần nhưng vẫn có sự chuyển tiếp mượt mà hơn so với lọc lý tưởng.

 **Lọc thông thấp Gaussian:** Ảnh mờ nhưng ít nhiễu hơn cả hai bộ lọc trên, vì Gaussian có sự chuyển tiếp mềm mại nhất giữa các tần số.

* Giải thích:

 **Bộ lọc lý tưởng** cắt bỏ hoàn toàn tần số cao một cách đột ngột, tạo ra biên sắc nét trong miền tần số, dẫn đến vân sáng tối xen kẻ trong ảnh không gian

 **Bộ lọc Butterworth** giảm dần tần số cao theo cấp số mũ, giúp làm mịn biên nhưng vẫn có một chút hiệu ứng này khi bậc Butterworth thấp.

 **Bộ lọc Gaussian** có dạng giảm tần số theo phân bố chuẩn (Gaussian)và tạo hiệu ứng làm mờ tự nhiên nhất.

**2.** Thực hiện phương pháp demosaicing được mô tả trong link sau

<https://docs.baslerweb.com/visualapplets/files/manuals/content/examples%20Bayer%20Laroche%20Filter.html>

Mô tả ngắn gọn phương pháp demosaicing trên. Phương pháp này có gì giống/khác so với phương pháp bilinear và Alleyssons?

*[Giải thích]*

**Mô tả ngắn ngọn phương pháp democaicing trên:**

Phương pháp Laroche phụ thuộc vào hướng cạnh để phục hồi ảnh màu từ dữ liệu Bayer. Việc phục hồi này được thực hiện đầu tiên với kênh màu xanh lá, khi kênh màu xanh lá được phục hồi đầy đủ thì tiếp tục phục hồi đến kênh màu xanh và đỏ. Việc phục hồi này phụ thuộc vào hướng cạnh để giảm sai số khi nội suy và giữ được độ sắc nét của cạnh bằng cách vận dụng phương pháp tái tạo dưa trên giả thuyết khác biệt màu sắc không đổi

- Điểm giống nhau :

- Cả ba phương pháp đều là các thuật toán nội suy màu, tái tạo màu dựa trên dữ liệu Bayer

- Dựa trên giá trị của pixel lân cận để điều chỉnh màu cho từng pixel

- Áp dụng trên từng kênh màu để khôi phục lại ảnh màu đầy đủ

- Điểm khác nhau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí** | **Bilinear** | **Laroche** | **Alleyson** |
| **Nguyên lý** | Trung bình các điểm lân cận | Dựa trên hướng cạnh | Tính toán hướng cạnh tối ưu bằng mô hình thống kê |
| **Chất lượng ảnh** | Kém, dễ mờ | Tốt, giữ được cạnh | Tốt nhất,sắc nét nhất |
| **Tạo nhiễu màu** | Cao | Trung Bình | Thấp nhất |
| **Tốc độ xử lý** | Nhanh | Trung bình | Chậm do độ phức tạp cao |

Minh họa kết quả của phương pháp trên. Tính SSIM và PSNR với phương pháp trên cho tập ảnh Kodak. So sánh với phương pháp bilinear và Alleyssons.

*[Code]*

I = imread('ngonhaidang.jpeg'); % ??c ?nh

I = double(I); % Chuyen sang kieu double

I\_org=I;

[h, w, c] = size(I);

I\_r=I(:,:,1);

I\_g=I(:,:,2);

I\_b=I(:,:,3);

%tao anh moi them vien ben ngoai bang 0

I\_padded = zeros(h+4, w+4, c); % anh moi co kich thuoc lon hon anh cu 4 cot va 4 hang

% chen anh goc vao giua anh moi

I\_padded(3:h+2, 3:w+2, :) = I;

I\_R = I\_padded(:,:,1);

I\_G = I\_padded(:,:,2);

I\_B = I\_padded(:,:,3);

[H\_img, W, C] = size(I\_padded); % Luu lai kich thuoc anh moi

lr = zeros(size(I\_padded));

%bayer CFA

lr(3:2:H\_img-2, 3:2:W-2, 1) = I\_R(3:2:H\_img-2, 3:2:W-2);

lr(4:2:H\_img-2, 4:2:W-2, 3) = I\_B(4:2:H\_img-2, 4:2:W-2);

lr(4:2:H\_img-2, 3:2:W-2, 2) = I\_G(4:2:H\_img-2, 3:2:W-2);

lr(3:2:H\_img-2, 4:2:W-2, 2) = I\_G(3:2:H\_img-2, 4:2:W-2);

%noi suy mau G tai pixel R

for i=3:2:H\_img-2

for j=3:2:W-2

dentaH1=(lr(i-2,j,1)+ lr(i+2,j,1))/2;

dentaV1=(lr(i,j-2,1)+ lr(i,j+2,1))/2;

if (dentaH1<dentaV1)

lr(i,j,2)=(lr(i-1,j,2)+lr(i+1,j,2))/2;

elseif(dentaH1>dentaV1)

lr(i,j,2)=(lr(i,j-1,2)+lr(i,j+1,2))/2;

else

lr(i,j,2)=(lr(i,j-1,2)+lr(i,j+1,2)+lr(i-1,j,2)+lr(i+1,j,2))/4;

end

end

end

%noi suy mau G tai vi tri pixel B

for i=4:2:H\_img-2

for j=4:2:W-2

dentaH1=(lr(i-2,j,1)+ lr(i+2,j,1))/2;

dentaV1=(lr(i,j-2,1)+ lr(i,j+2,1))/2;

if (dentaH1<dentaV1)

lr(i,j,2)=(lr(i-1,j,2)+lr(i+1,j,2))/2;

elseif(dentaH1>dentaV1)

lr(i,j,2)=(lr(i,j-1,2)+lr(i,j+1,2))/2;

else

lr(i,j,2)=(lr(i,j-1,2)+lr(i,j+1,2)+lr(i-1,j,2)+lr(i+1,j,2))/4;

end

end

end

%noi suy mau B tai vi tri pixel R

for i=3:2:H\_img-2

for j=3:2:W-2

lr(i,j,3)=(lr(i-1,j-1,3)-lr(i-1,j-1,2)+lr(i+1,j-1,3)-lr(i+1,j-1,2)+lr(i+1,j+1,3)-lr(i+1,j+1,2)+lr(i+1,j-1,3)-lr(i+1,j-1,2))/4+lr(i,j,2);

end

end

%noi suy mau R tai vi tri pixel B

for i=4:2:H\_img-2

for j=4:2:W-2

lr(i,j,1)=(lr(i-1,j-1,1)-lr(i-1,j-1,2)+lr(i+1,j-1,1)-lr(i+1,j-1,2)+lr(i+1,j+1,1)-lr(i+1,j+1,2)+lr(i+1,j-1,1)-lr(i+1,j-1,2))/4+lr(i,j,2);

end

end

%noi suy mau B va R tai vi tri pixel G

for i=4:2:H\_img -2

for j=3:2:W-2

lr(i,j,1)=(lr(i,j-1,1)-lr(i,j-1,2)+lr(i,j+1,1)-lr(i,j+1,2))/2+lr(i,j,2);

lr(i,j,3)=(lr(i,j-1,3)-lr(i,j-1,2)+lr(i,j+1,3)-lr(i,j+1,2))/2+lr(i,j,2);

end

end

%noi suy mau B va R tai vi tri pixel G

for i= 3:2:H\_img-2

for j=4:2:W-2

lr(i,j,1)=(lr(i,j-1,1)-lr(i,j-1,2)+lr(i,j+1,1)-lr(i,j+1,2))/2+lr(i,j,2);

lr(i,j,3)=(lr(i,j-1,3)-lr(i,j-1,2)+lr(i,j+1,3)-lr(i,j+1,2))/2+lr(i,j,2);

end

end

lr\_cropped = lr(3:h+2, 3:w+2, :);

subplot(121);imshow(uint8(I\_org));title('anh goc');

subplot(122);imshow(uint8(lr\_cropped));title('anh sau khi dung laroche');

mse = (double(I\_org)-double(lr\_cropped)).^2 ;

mse = mean(mse(:)) ;

PSNR = 10\*log10(255.^2/mse);

disp(PSNR);

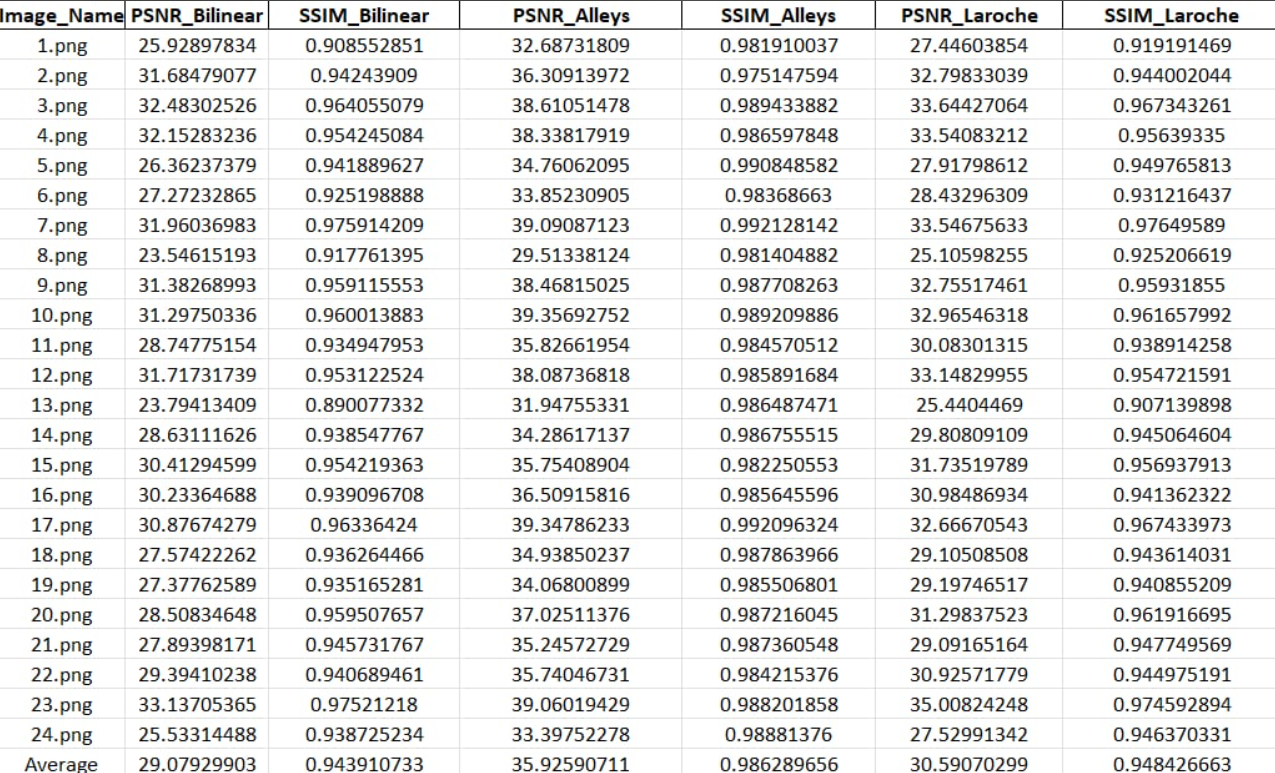
ssim\_val = ssim(uint8(lr\_cropped), uint8(I\_org));

disp(ssim\_val);

*[Hình/bảng]*



Bảng so sánh hệ số PSNR và SSIM của tập ảnh Kodak của 3 phương pháp: Bilinear, Alleyssons và Laroche:

**

*[Giải thích]*

Nhận xét kết quả khi so sánh với hai phương pháp còn lại là Bilinear và Alleysson

* PSNR:
  + Phương pháp Alleyson có chỉ số PSNR lớn nhất tức là ảnh tái tạo gần với ảnh gốc nhất, ít nhiễu hơn
  + Laroche có chỉ số PSNR thấp hơn Alleys nhưng vẫn cao hơn Bilinear tức là cải thiện hơn Bilinear nhưng không tối ưu bằng Alleyssons
  + Bilinear có PSNR thấp nhất tức là ảnh tái tạo mờ và có nhiều nhiễu do không tái tạo theo hướng cạnh
* SSIM:
  + Alleys có chỉ số PSNR cao nhất nên SSIM của nó cũng gần với ảnh gốc nhất tức là gần với 1 tương đương với giống 100 %
  + Laroche thì có chỉ số PSNR thấp hơn Alleys nên là chỉ số SSIM của nó cũng chỉ nằm ở tầm trung khoảng 95% đến 97% không tối ưu bằng Alleys
  + Bilinear có chỉ số PSNR thấp nhất nên SSIM của nó cũng thấp nhất vì ảnh mờ và chứa nhiều nhiễu

3. Xét bộ lọc FL trong phương pháp demosaicing của Alleyssons. Hiển thị phổ biên độ của bộ lọc này theo tần số chuẩn hóa trong khoảng [-0.5,0.5). Giải thích vai trò của bộ lọc FL trong phương pháp Alleyssons.

*[Code]*

import cv2

import matplotlib.pyplot as plt

import numpy as np

from scipy.signal import convolve2d

FL = np.array ([[-2, 3, -6, 3, -2],

                [ 3, 4, 2, 4, 3],

                [-6, 2, 48, 2, -6],

                [3, 4, 2, 4, 3],

                [-2, 3, -6, 3, -2]]) / 64

fft\_FL = np.fft.fft2(FL)

shift\_fft\_FL = np.fft.fftshift(fft\_FL)

magnitude\_FL = np.abs(shift\_fft\_FL)

log\_magnitude = np.log(1 + magnitude\_FL)

plt.figure(figsize=(12, 10))

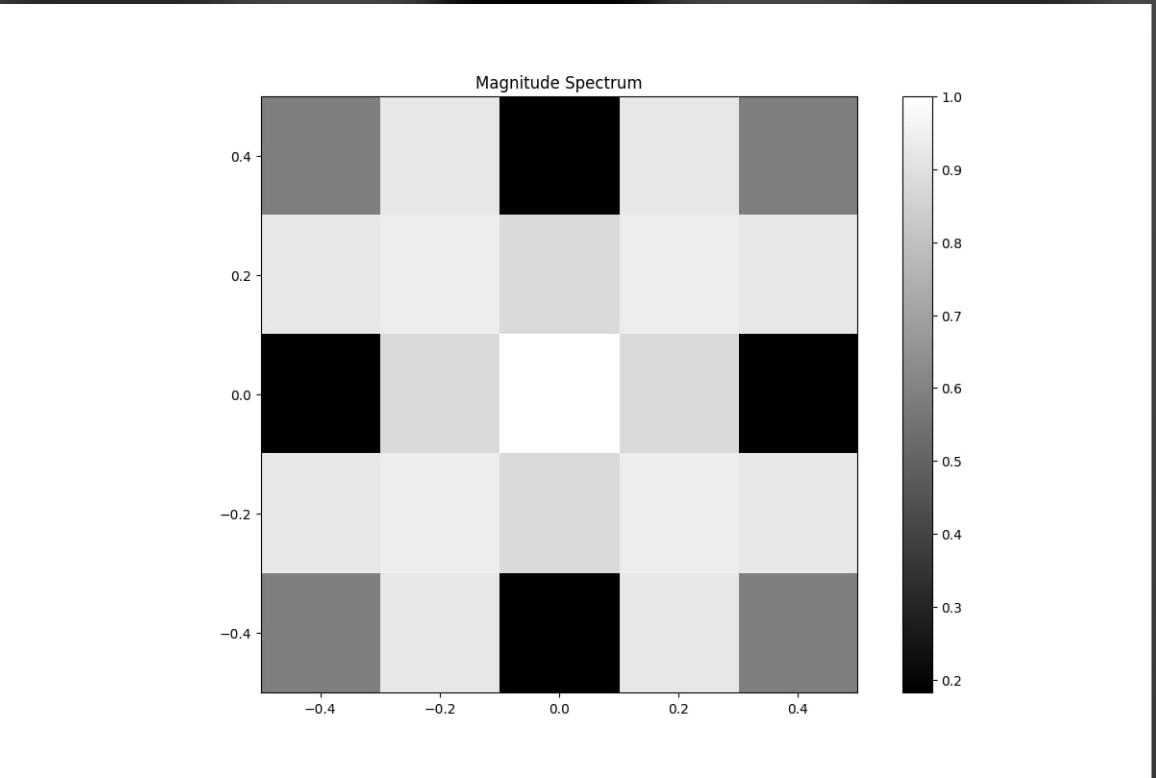
plt.imshow(magnitude\_FL, cmap='gray' , extent=[-0.5, 0.5, -0.5, 0.5])

plt.title("Magnitude Spectrum")

plt.colorbar()

plt.show()

*[Hình/bảng]*

**

*[Giải thích]*

Vai trò của bộ lọc trong phương pháp Alleyssons:

* Dùng để ước lượng độ sáng của từng pixel
* Bộ lọc này đóng vai trò như bộ lọc thông thấp, giúp lọc nhiễu do nội suy tạo ra, loại bỏ những thành phần tần số cao không cần thiết