TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN



**Bài tập lớn**

XỬ LÝ ẢNH

Đề số: 27

|  |  |
| --- | --- |
| GIẢNG VIÊN HƯỚNG DẪN : | VŨ HUẤN |
| SINH VIÊN THỰC HIỆN : | HOÀNG CAO LONG-181200401  TRẦN CHÍ THÀNH-181210042 |

Hà Nội, Ngày … Tháng … Năm ….

**I. Câu hỏi thực hành**

1. Tạo một ảnh có kích thức 330x530 gồm nền màu đen và một đĩa tròn có bán kính 30 pixels màu trắng . Đặt hình vào góc dưới bên trái ảnh.

Code:

X = zeros(330,530);

X(295:305,1:60)=255;

X(291:294,2:59)=255;

X(287:290,3:58)=255;

X(285:286,4:57)=255;

X(283:284,5:56)=255;

X(282,6:55)=255;

X(280:281,7:54)=255;

X(279,8:53)=255;

X(278,9:52)=255;

X(277,10:51)=255;

X(276,11:50)=255;

X(275,13:48)=255;

X(274,14:47)=255;

X(273,16:45)=255;

X(272,18:43)=255;

X(271,21:42)=255;

X(270,25:38)=255;

X(306:309,2:59)=255;

X(310:312,3:58)=255;

X(313:314,4:57)=255;

X(315:316,5:56)=255;

X(317,6:55)=255;

X(318:319,7:54)=255;

X(320,8:53)=255;

X(321,9:52)=255;

X(322,10:51)=255;

X(323,11:50)=255;

X(324,13:48)=255;

X(325,14:47)=255;

X(326,16:45)=255;

X(327,18:43)=255;

X(328,21:42)=255;

X(329,25:38)=255;

X(270:330,25:35)=255;

imshow(X);

Ảnh:



2. Mở ảnh image27.jpg và chuyển ảnh thành dạng ảnh độ xám 8-bit và hiển thị ảnh.

Code:

rgb = imread('image27.jpg');

image = rgb2gray(rgb);

imshow(image);

Ảnh:



3. Sử dụng phương pháp cắt bit-plane và sử dụng bit-plane 4-5-6-7 để khôi phục ảnh. Hiển thị lược đồ xám của ảnh ban đầu và ảnh sau khi cắt bit-plane.

Code:

divide = bitPlaneSlicing(image);

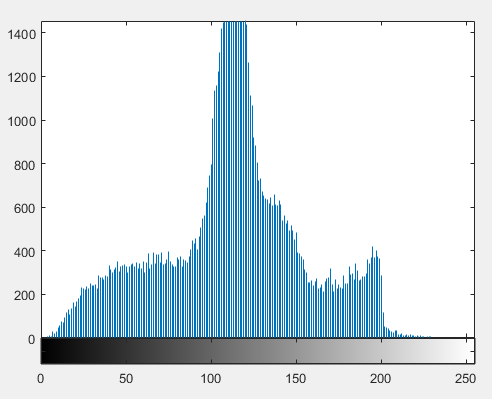
restore\_4\_5\_6\_7 = divide(:,:,7)\*64 + divide(:,:,6)\*32 + divide(:,:,5)\*16 + divide(:,:,4)\*8;

imhist(image);

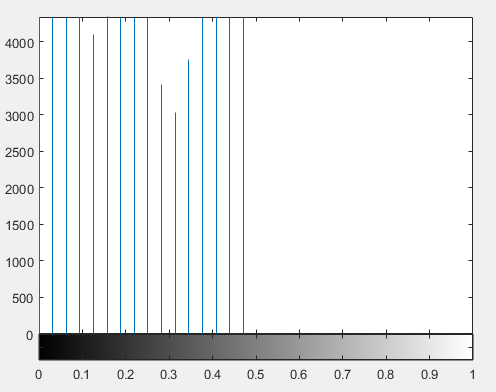
imhist(restore\_4\_5\_6\_7/255);

Ảnh:

* Lược đồ xám ảnh ban đầu:



* Lược đồ xám ảnh sau khi cắt bit-plane:



4. Hiển thị phổ tần số của ảnh trên thang tuyến tính và trên thang log. Đưa ra nhận xét.

Code:

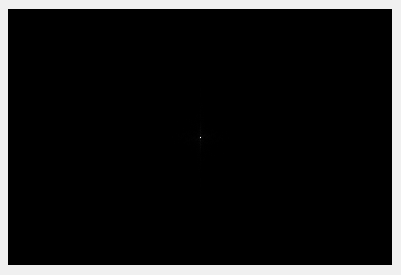
Fourier = fftshift(fft2(image));

imshow(mat2gray(abs(Fourier)));

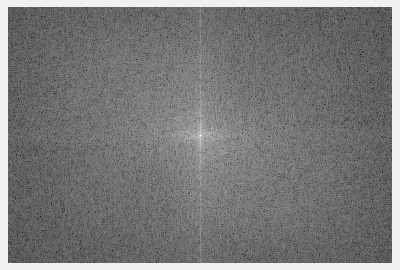
imshow(mat2gray(log(abs(Fourier))));

Ảnh:

* Ảnh trên thang tuyến tính:



* Ảnh trên thang log:



5. Chuẩn hoá ảnh và đưa vào ảnh một nhiễu Gausse có phương sai 0.15. Hiển thị ảnh nhiễu và nhận xét.

Code:

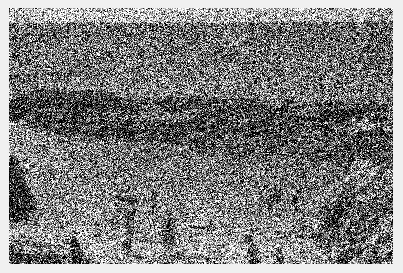
convert\_double = double(image);

standardized = convert\_double/255;

Gausse = imnoise(standardized, 'gaussian', 0, 0.15);

imshow(Gausse);

Ảnh:



Nhận xét :

* Ảnh bị nhiễu nặng

6. Áp dụng bộ trung vị 5x5 vào ảnh nhiễu. Ảnh có bớt nhiễu không? Nếu không thì nên áp dụng bộ lọc nào? Tính PSNR và đưa ra nhận xét.

Code:

Median = medfilt2(Gausse, [5 5]);

PSNR\_Median = psnr(Median,Gausse) = ~ 10,0395;

h = fspecial('average',[5 5]);

Mean = imfilter(Gausse,h,'replicate');

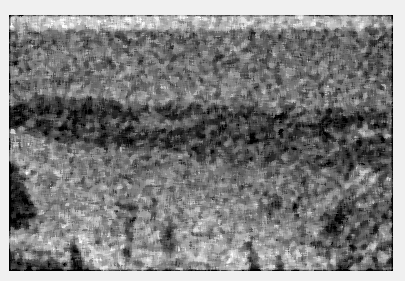
PSNR\_Mean = psnr(Mean,Gausse) = ~ 10,2892;

imshow(Median);

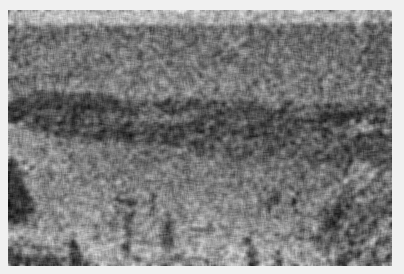
imshow(Mean);

Ảnh:

* Ảnh qua bộ lọc trung vị:



* Ảnh qua bộ lọc trung bình:



Nhận xét:

* Ảnh qua bộ lọc trung vị không lọc hết được , vẫn sót lại 1 vài điểm trắng
* Ảnh qua bộ lọc trung bình có thể lọc được hết , tuy nhiên ảnh bị mở
* Chọn bộ lọc trung bình

7. Chọn ra một loại đường viền mà sinh viên muốn áp dụng với ảnh. Thử áp dụng các kỹ thuật làm nổi đường viền (lọc gradient, lọc Sobel, lọc Laplace, lọc LOG) và chọn ra bộ lọc tốt nhất đối với yêu cầu trên. Hiển thị ảnh sau lọc.

Code:

h=fspecial('log',[5,5],1);

LOG=imfilter(standardized,h5);

imshow(mat2gray(LOG));

Ảnh:

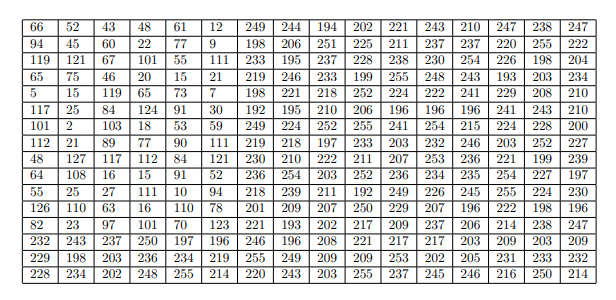


Nhận xét :

* Bộ lọc tốt nhất là bộ lọc log vì hiển thị được khá nhiều đường viền

**II. Câu hỏi tính toán**

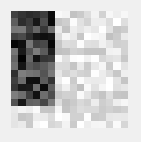
Cho một ma trận ảnh như sau:



1. Sử dụng phương pháp chọn ngưỡng tự động Otsu để chia ảnh thành hai phần nền và đối tượng. Hiển thị kết quả sau khi phân ngưỡng.

Ảnh:

* Ảnh ban đầu



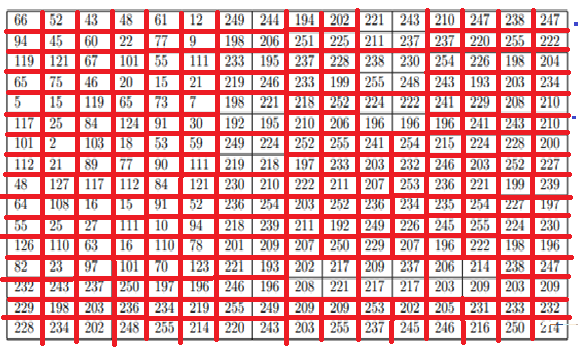
* Ảnh sau khi phân ngưỡng



2. Mô tả từng bước của thuật toán Split and Merge trên ảnh đã cho. Sinh viên tự chọn các thông số của thuật toán Split and Merge.

Giá trị độ xám lớn nhất và nhỏ nhất trong vùng không quá 32

Split:



Merge:

