BÁO CÁO BÀI TẬP VỀ NHÀ

MÔN HỌC: XỬ LÝ ẢNH

SV: Phạm Bá Tùng – B15DCDT221

Nhóm 1 – STT: 53

1.

I.yêu cầu

thực hiện số hóa các điểm để có được mặt phẳng nhìn từ trên xuống

II.cách làm

1.chọn detalX , deltaly

2. chọn xmin , xmax , ymin ymax

3.sử dụng công thức đã cho z = x^2 + y^2 tính ra giá trị các của mật phảng nhìn từ Z xuống

III.kết quả

đúng với khi tính bằng tay

IV.code chính

while i <= Xmax

i += detalX;

j = Ymin;

while j <= Ymax

j += detalY;

printf("%6.1f" ,(i^2 + j^2) )

endwhile;

printf("\n");

endwhile;

………………………………….

2 yêu cầu

sử dụng hệ màu hsv phát hiện tín hiệu đèn giao thông

cách làm

1. chuyển đổi không gian màu rgb > hsv

2.xác định màu của đèn đỏ ( ở mã hsv)

3. duyệt ảnh > các điểm có mã màu gần đúng với màu đỏ (hsv) đã tìm được ở trên

4. imshow kết quả kết quả

gần chính xác . do xung quanh màu đỏ của đèn giao thông còn có nhiễu . màu đỏ ko đều

code chính

for i=1:size(I,1)

for j=1:size(I,2)

if( (I(i,j) > I(14,221) - 20) && I(i,j) < I(14,221)+20)

Ir(i,j,:) = [255,255,255];

endif;

endfor;

endfor;

/…………………………..

3

yêu cầu

code thuật toán khếch tán lỗi 1D và 2D

cách làm

theo đúng lý thuyết

1. dựa trên ngưỡng >> xác định được lỗi sau khi phân ngưỡng

2. lỗi sẽ được cộng theo từng quy tắc ( 1D hoặc 2D )

đối với 2D cần có ma trận khuếch tán lỗi

3. kết quả ghi ra file resultThresh.txt

kết quả

chạy đúng so với tính tay

code chính

2d

res1 = addError(res1,i,j,err,matrixhuechTanLoi);

if( res1(i,j) > thresh)

err = 255 - res1(i,j);

res1(i,j) = 255;

else

err = -res1(i,j);

res1(i,j) = 0;

endif;

1D

res1(i,j) += err;

if( res1(i,j) > thresh)

err = 255 - res1(i,j);

res1(i,j) = 255;

else

err = -res1(i,j);

res1(i,j) = 0;

endif;

cách chạy : gõ tên file

.4……...

bai 4

yêu cầu

+cần bằng lược đồ xám , xử lý bó cụm

+ so sánh kết quả với hàm thư viện

cách làm

1. tính xác suất lưu vào pElement

2. áp dụng công thức đã có page7 dip3

ý b ;

áp dụng công thức đã có page9 dip3

kết quả

không giống với hàm thư viện lắm

code chính

ý a

for i= 1:Lmax+1

for j =1:i

Ihat(i)+=Lmax\*pElement(j);

endfor ;

endfor;

Ihat = round(Ihat);

ý b

Ihat = floor(I./Bsize) \*Bsize;

cách chạy :

%ý a cân bằng lược đồ xám với ảnh I4

bai4a(I4);

%ý b chọn ảnh I4 , với kích thước Bsize = 50 ; tt bó cụm

bai4b(I4,50);

………………………..

5

……yêu cầu

tính giá trị trung bình và phương sai

cách làm

sử dụng công thức đã có kết quả

chạy đúng với cách làm nhanh của octave ( xử lý phép tính trên ma trận luôn không

dùng vòng lặp )

code chính

for i = 1:widthI

for j = 1:heighI

sumI += Isource(i,j);

endfor;

endfor;

tb = sumI / (widthI \*heighI);

% tính

for i= 1:widthI

for j = 1:heighI

sum2 += (Isource(i,j) - tb)^2;

endfor;

endfor;

ow = sum2 / (widthI\*heighI);

sử dụng

% load module

% chạy code test với tham số ảnh đầu vào là I5;

codeTestb5(I5) ;

…………….

6.

…

yêu cầu

thực hiện bộ lọc trung bình , trung vị , max , min cho ảnh

cách làm

1. lấy cửa sổ mặt nạ nhân tương ứng về vị trí với các phần tử cạnh phần tử xét

2. tính toán dựa trên công thức đối với từng bộ lọc

3. show kết quả bằng imshow ,so sánh kết quả với imfilter;

kết quả

chạy gần như giống với hàm có sẵn . có thể do cách làm tròn

code chính

%trung bình

mWindow = getWindow(Isource , x , y , win);

s = sum(sum(win));

Average = uint8(sum(sum(mWindow))/s);

%trung vị

if(mod(heightTrungVi , 2) == 1)

giua1 = round(heightTrungVi/2);

giua2 = giua1;

else

giua1 = floor(heightTrungVi/2);

giua2 = ceil(heightTrungVi/2) ;

endif;

trungVi = (trungViMatrix(giua1) + trungViMatrix(giua2 ))/2;

%max

result4(i,j) = getMaxFilterAtPoint(I,i,j,win);

%min

result4(i,j) = getMinFilterAtPoint(I,i,j,win);

cách chạy

% load module bài 6

%chạy function test

codeTestb6(I6);

……

7.

yêu cầu

tính module gradien và góc pha của ảnh khi áp dụng mặt nạ

SOBEL , PREWITT , ROBERT

cách làm

1. tạo của sổ ( sobel , prewitt , robert);

2. ốp mặt nạ lên matrix ảnh

3. tính gradiant và góc pha dựa trên công thức đã có

kết quả

giống với hàm imfilter của thư viện

code chính

for i = 1:widthI

for j = 1 : heightI

Average(i,j) = getMatrixWithWinAtPoint(Isource , i,j, win);

endfor;

endfor;

cách chạy

% load module

% chạy function test với I7 là ma trận đầu vào

codeTestb7(I7);

% chạy code test của library với I7 là ảnh đầu vào

codeTestLibraryb7(I7) ;

…………………

9.

yêu cầu

tính SNR và MSE , so sánh kết quả với hàm thư viện

cách làm

1. load bài 6 đã làm để lấy function getAverageMatrix

run ../

2. sử dụng công thức đã biết để tính snr và mse

kết quả

chạy giống vơi thư viện

code chính

IAverage = getAverageMatrix(I, w1);

SNR = sum(sum(IAverage.^2))/sum(sum((I-IAverage).^2))

MSE = sum(sum((I-IAverage).^2))/size(I,1)/size(I,2)

cách chạy

%load tất cả function trong bài 9

%chạy code sử dụng thư viện

codeTestLibrary9(I9);

% chạy code tự code

codeTestb9(I9);

……

10…

yêu cầu

sử dụng hàm thư viện thực hiện phân ngưỡng ảnh theo biên

cách làm

1 dùng edge để phát hiện biên

2 lam mảnh bien

kết quả

chính xác

code chính

Igray = rgb2gray(I);

cFilter = edge(Igray , "canny");

thinEdgeImage = bwmorph(cFilter, 'Skel', inf);