**Experiment 4 image restoration and image segmentation**

1. **The purpose of the experiment**
2. Familiarity with and mastering the use of the MATLAB image processing toolbox
3. Understand and master commonly used image restoration and segmentation techniques.
4. **Experimental environment**

MATLAB 6.5 and above, WIN XP or WIN2000 computer

1. **Relevant knowledge**
2. deconvwnr Wiener filter

usage:

J = deconvwnr(I,PSF,NSR)

Use Wiener filter algorithm to restore image I and return to image J. I is an N-dimensional array,PSF is the convolution of the point spread function.NSP is the noise-to-signal power ratio of additive noise

Eg:

I = im2double(imread('cameraman.tif'));

imshow(I);

title('Original Image');

% Simulate motion blur

LEN = 21;

THETA = 11;

PSF = fspecial('motion', LEN, THETA);

blurred = imfilter(I, PSF, 'conv', 'circular');

figure, imshow(blurred)

% restore image

wnr2 = deconvwnr(blurred\_noisy, PSF);

figure, imshow(wnr2)

title('Restoration of Blurred')

1. edge

Detects edges in a grayscale or binary image and returns a binary image where 1 pixel is a detected edge and 0 pixel is a non-edge.

Usage:

BW = edge(I,'sobel',thresh,direction)

I is the detection object; The edge detection operator can be used sobel , roberts , prewitt , zerocross , log , canny ;Thresh specifies the threshold, ignores all edges smaller than the threshold during detection, and automatically selects the threshold by default; direction, in the specified direction, uses the specified operator to perform edge detection

Horizontal,vertical or both.

eg：

I = imread('circuit.tif');

BW1 = edge(I,'prewitt');

imshow(BW1);

1. strel Create Morphological Structural Elements

usage：

SE = STREL('arbitrary',NHOOD,HEIGHT)

Creates a non-planar structured element that specifies a domain. HEIGHT is a matrix of the same size as NHOOD that specifies the height of any non-zero element in NHOOD.

SE = STREL('ball',R,H,N) creates a spatial ellipsoid-shaped structuring element with a radius R in the X-Y plane and a height H. R must be a non-negative integer and H is a real number. N must be a non-negative even number. When N>0 this spherical structuring element is approximated by a series of spatial line segment structuring elements.

SE = STREL('diamond',R) creates a structuring element in the shape of a flat diamond of the specified size R. R is the distance from the origin of the structuring element to its point, and must be a non-negative integer. SE = STREL('disk',R,N) creates a flat disk-shaped structuring element of the specified radius R. Here R must be a non-negative integer. N must be 0, 4, 6, 8. When N is greater than 0, the disk-shaped structuring element is approximated by a set of N (or N+2) periodic line structuring elements. When N is equal to 0, no approximation is used, that is, all pixels of the structuring element are composed of pixels whose distance to the central pixel is less than or equal to R. N can be ignored, in this case the default value is 4. Note: Morphological operations are faster in the case of N>0 than in the case of N=0.

Eg:

se1 = strel('square',11) % 11 times 11 square

1. imerode Corroded image

Usage：IM2 = imerode(IM,SE)

Corrodes grayscale, binary, or compressed binary image IM , returning corroded image IM2 . The parameter SE is a structure element body or an array of structure element bodies returned by the function strel.

Eg: Corrodes a binary image using a disc-shaped structuring element.

originalBW = imread('circles.png');

se = strel('disk',11);

erodedBW = imerode(originalBW,se); imshow(originalBW), figure, imshow(erodedBW)

1. Imdilate inflated image

Usage:IM2 = imdilate(IM, SE)

Dilated grayscale, binary, or packed binary image IM , returns dilated image M2 . The variable SE is a struct element or an array of struct elements, which is returned by the strel function.

Eg:Dilate grayscale images with a running structuring element

I = imread('cameraman.tif');

se = strel('ball',5,5);

I2 = imdilate(I,se);

imshow(I), title('Original')

figure, imshow(I2), title('Dilated')

1. **Experimental steps**

1.Generate motion blurred images, apply Wiener filtering for image restoration, and display the results.

i=imread('f:\1.jpg')

I=rgb2gray(s)

I = im2double(I); % Simulate motion blur

LEN = 21; THETA = 11;

PSF = fspecial('motion', LEN, THETA);

blurred = imfilter(I, PSF , 'conv', 'circular'); % restore image

wnr2 = deconvwnr(blurred, PSF);

subplot(1,2,1);imshow(blurred);title('motion blurred image')

subplot(1,2,2);imshow(wnr2);title('restore image')

2.Use three different operators to detect image edges and display the results.

i=imread('f:\1.jpg')

BW1 = edge(I,'prewitt'); BW2 = edge(I,'zerocross');

BW3 = edge(I,'canny');

subplot(2,2,1);imshow(i);title(' original image ')

subplot(2,2,2);imshow(BW1);title(' prewitt edge map ')

subplot(2,2,3);imshow(BW2);title('zerocross edge map ')

subplot(2,2,4);imshow(BW3);title(' canny edge map ')

3.Perform the dilation and erosion operations of the square template 3\*3 and 5\*5 respectively on the binary image, and display the results.

a=imread('f:\1.jpg')

i = rgb2gray(a)

I = im2bw(a,0.5) se3 = strel('disk',3); erodedBW1 = imerode(I,se3);

se4 = strel('disk',5); erodedBW2 = imerode(I,se4);

se1 = strel('ball',3,3);

I1 = imdilate(a,se1);

se2 = strel('ball',5,5);

I2 = imdilate(a,se2);

subplot(2,2,1);imshow(I1);title('3\*3 inflated image ')

subplot(2,2,2);imshow(I2);title('5\*5 inflated image ')

subplot(2,2,3);imshow(erodedBW1);title('3\*3 Corroded image ')

subplot(2,2,4);imshow(erodedBW2);title('5\*5 Corroded image ')