Digital Image Processing

Presented by:

Dr. Moe Moe Myint

Information Technology Department

Technological University (Kyaukse), Myanmar

Gmail moemoemyint@moemyanmar.ml

slideshare www.slideshare.net/MoeMoeMyint

Dr. Moe Moe Myint
Information Technology Department
Technological University (Kyaukse)

Linear Filtering (Lab 8)

The purpose of image enhancement is to improve the visual appearance of an image for human or Computer analysis. Filtering (including Fourier filtering) is one of the techniques used for image Enhancement to filtering noise, to emphasize the low, high or directional spatial frequency components, etc.

Objectives

- To use 2-D median filtering
- To use 2-D filtering of multidimensional images

Required Equipment

- Computers with MATLAB software and Projector

Practical Procedures

- Use the *fspecial* command
- Use the *imfilter* command

fspecial

- Create predefined 2-D filter

<u>Syntax</u>

- h = fspecial(type)
- h = fspecial(type, parameters)

<u>Description</u>

h = fspecial(type) creates a two-dimensional filter h of the specified type. fspecial returns h as a correlation kernel, which is the appropriate form to use with imfilter. type is a string having one of these values.

Examples

```
I = imread('cameraman.tif');
subplot(2,2,1);
imshow(I); title('Original Image');
H = fspecial('motion',20,45); %Approximates the linear motion of a camera
MotionBlur = imfilter(I,H,'replicate'); %Input array values outside the bounds of the array are assumed to equal the nearest array border value
subplot(2,2,2);
imshow(MotionBlur);title('Motion Blurred Image');
H = fspecial('disk',10); %Circular averaging filter (pillbox)
blurred = imfilter(I,H,'replicate');
subplot(2,2,3);
imshow(blurred); title('Blurred Image');
H = fspecial('unsharp'); %unsharp contrast enhancement filter
sharpened = imfilter(I,H,'replicate');
subplot(2,2,4);
imshow(sharpened); title('Sharpened Image');
```

imfilter

- N-D filtering of multidimensional images

Syntax

B = imfilter(A, H)

Description

- B = imfilter(A, H) filters the multidimensional array A with the multidimensional filter H. The array A can be logical or a nonsparse numeric array of any class and dimension. The result B has the same size and class as A.

Examples

originalRGB = imread('peppers.png');

% Read a color image into the workspace and view it.

imshow(originalRGB)

h = fspecial('motion', 50, 45); %Create a filter, h, that can be used to approximate linear camera motion.

filteredRGB = imfilter(originalRGB, h); % Apply the filter, using imfilter, to the image originalRGB to create a new image, filteredRGB.

figure, imshow(filteredRGB)

boundaryReplicateRGB = imfilter(originalRGB, h, 'replicate');

% Specify the replicate boundary option.

figure, imshow(boundaryReplicateRGB)

Example

```
I = imread('moon.tif');
h = fspecial('unsharp');
I2 = imfilter(I,h);
imshow(I), title('Original Image')
figure, imshow(I2), title('Filtered Image')
```

Example

```
clc,clear all,close all;
I = im2double(imread('cameraman.tif'));
% imshow(I);title('Original Image (courtesy of MIT)');
LEN = 21; THETA = 11;
PSF = fspecial('motion', LEN, THETA);
blurred = imfilter(I, PSF, 'conv', 'circular');
figure,imshow(blurred);title('Blurred Image');
wnr1 = deconvwnr(blurred, PSF, 0);%Use deconvwnr to Restore an Image
figure,imshow(wnr1);title('Restored Image');
noise_mean = 0; noise_var = 0.0001;
blurred_noisy = imnoise(blurred, 'gaussian', noise_mean, noise_var);
figure,imshow(blurred_noisy);title('Simulate Blur and Noise')
wnr2 = deconvwnr(blurred_noisy, PSF, 0);
% figure,imshow(wnr2);title('Restoration of Blurred, Noisy Image Using NSR = 0')
signal_var = var(I(:)); wnr3 = deconvwnr(blurred_noisy, PSF, noise_var / signal_var);
figure,imshow(wnr3);title('Restoration of Blurred, Noisy Image Using Estimated NSR');
```

Questions?