

LAB LESSON FIVE
SPATIAL LINEAR FILTERING
Dr. Nursuriati Jamil

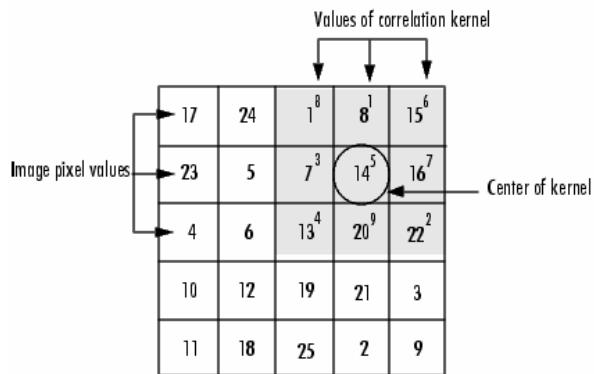
Linear Filtering

A linear filter computes each output pixel value according to a linear combination of the input pixel's neighborhood. The basics of linear filtering are done through correlation and convolution.

Correlation

In correlation, the value of an output pixel is computed as a weighted sum of neighboring pixels. Suppose a 3x 3 filter is defined as:

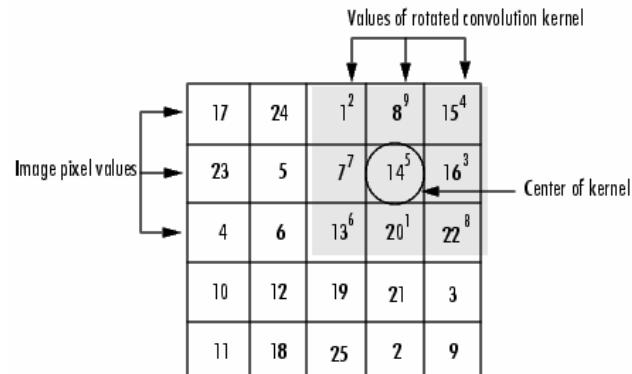
8	1	6
3	5	7
4	9	2



Correlation filtering: $(1 \times 8) + (8 \times 1) + (15 \times 6) + (7 \times 3) + (14 \times 5) + (16 \times 7) + (13 \times 4) + (20 \times 9) + 22(2) = 585$

Convolution

In convolution, the value of an output pixel is also computed as a weighted sum of neighboring pixels. The difference is that the matrix of weights (filter) is rotated 180°.



Correlation filtering: $(1 \times 2) + (8 \times 9) + (15 \times 4) + (7 \times 7) + (14 \times 5) + (16 \times 3) + (13 \times 6) + (20 \times 1) + 22(8) = 575$

Performing Filtering in MATLAB

```
B = imfilter(A,H)
B = imfilter(A,H,option1,option2,...)
```

Where

A – input image

H – correlation/convolution filter

Options - boundary options, output size option, correlation and convolution option

Average Filtering

One of the most basic examples of a linear filter is that of an averaging filter. It results in the smoothing of sharp features and thus in the reduction of intensity disparity of pixels whose intensities are much above or below those of the neighbors (Seul et al. 2001). This filter removes high spatial frequencies from an image and performs well on Gaussian noise (Ko & Lee 1991) but in the presence of salt-and-pepper noise, median filtering outperforms mean filtering.

An averaging filter is commonly used in reducing the graininess of an image.

Below is an example of the cameraman picture with some added graininess that is then removed using an averaging filter.

```
I = imread('cameraman.tif');
% addition of graininess
I_noise = imnoise(I, 'speckle', 0.01);

% create average filter
h = ones(3,3) / 9;
I2 = imfilter(I_noise,h);
imshow(I_noise), title('Original image');
figure, imshow(I2), title('Filtered image');
```

Median Filtering

Bovic and Munson (1983) suggested that median filter is best used to suppress salt-and-pepper noise. Calculation of the median filter begins by ordering the N pixels included in the filter window and setting the center pixel to the median intensity value (Metherall 2000). This has the effect of forcing points with distinct intensities to be similar to its neighbor, thus eliminating intensity spikes that appear isolated. Median filter is one of the most popular nonlinear filters (Bernstein 1987); it is computationally efficient and can preserves or reconstruct edge and other image details while removing noise. Median filter is one of the most popular nonlinear filters (Bernstein 1987); it is computationally efficient and can preserves or reconstruct edge and other image details while removing noise.

```
% Median filter
I = imread('eight.tif');
J = imnoise(I,'salt & pepper',0.02);
K = medfilt2(J);
imshow(J), figure, imshow(K)
```

Creating Linear Filters

Matlab provides a function to create two-dimensional linear filter.

```
fspecial(type, parameters)
```

Where:

h – two-dimensional filter

type – one of the specified special filter types: gaussian, sobel, prewitt, laplacian, log, motion, averaging (average), circular averaging (disk), and a contrast sharpening (unsharp) filter

parameters – particular to the type of filter chosen

Region of Interest Filtering

Matlab provides a function to create two-dimensional linear filter .

```
J = roifilt2(h, I, BW)
```

Where:

I - image

H – two-dimensional filter

BW – binary image the same size as I that defines an ROI used as a mask for filtering

```
I = imread('cameraman.tif');
BW = roipoly(I);
H = fspecial('average', [15, 15]);
J = roifilt2(H, I, BW);
figure, imshow(J);
```

References:

1. Ko, S.J. & Lee, Y.H. 1991. Center weighted median filters and their applications to image enhancement. *IEEE Transactions on Circuits and System* 38: 984-993.
2. Seul M., O'Gorman L. & Sammon M.J. 2000. *Practical algorithms for image analysis*. Cambridge: Cambridge University Press.
3. Bernstein, R. 1987. Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images. *IEEE Transactions on Circuits and Systems* 34(11):1275-1291.
4. Bovic, A.C., Huang T.S. & Munson, D.C. 1983. A generalization of median filtering using linear combination of order statistics. *IEEE ASSP* 31:1342-1349.
5. Metherall, G. 2000. Local segmentation of images. BSc. Thesis. School of Computer Science and Software Engineering, Monash University.

Assignment

1. Describe and explain the results of filtering 'pattern.tif' with average filter using different boundary options. (5 marks)
2. Remove noises in 'Noisy1.tif' and 'Noisy2.tif' images. (2 marks)
3. Remove the small blobs in 'Blobs.jpg'. (8 marks)