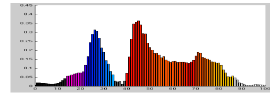


Dr. Qadri Hamarsheh

Digital Image Processing (750474)
Lecture 12

Fundamentals of spatial Filtering

Outline of the Lecture

- Introduction.
- Spatial Correlation and convolution.
- Vector Representation of linear filtering

Introduction

Filters in frequency domain:

- ***Lowpass filter*** that passes low frequencies: used for smoothing (blurring) on the image.
- ***Highpass filter*** that passes high frequencies: used for sharpening the image.
- ***Bandpass filter.***

Filters in spatial domain:

- Spatial filters used different **masks** (**kemels**, **templates** or **windows**).
- There is a ***one-to-one*** correspondence between ***linear*** spatial filters and filters in frequency domains.
- Spatial filters can be used for ***linear*** and ***nonlinear*** filtering. (Frequency domain filters just for linear filtering).
- The **mechanics** of spatial filtering spatial filters consists of:
  1. Neighbourhood (small rectangle).
  2. Predefined operation that is performed on the image pixel.

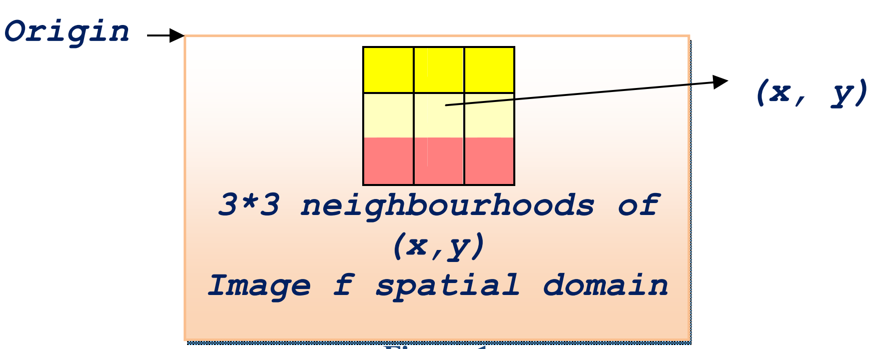
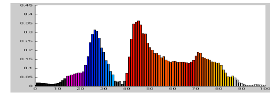


Figure 1

- ***Filtering*** creates new pixel with coordinates equal to the coordinates of the centre of the neighbourhood, and whose value is the result of the filtering operation.
  - If the operation performed on the image pixel is **linear**, then the filter is called a **linear spatial filter**, otherwise, the filter is **nonlinear**.
  - Figure 1 presents the mechanics of linear spatial filtering using a ***3*3 neighborhood***.
  - the ***response*** (output) $g(x,y)$ of the filter at any point (x,y) in the image is the sum of products of the filter coefficients and the image pixels values:



$$g(x,y) = w(-1,-1)f(x-1,y-1) + \; w(-1,0)f(x-1,y) + \\ \ldots + \; w\,(0,0)\;f(x,y) + \ldots + w(1,1)\;f(x+1,y+1)$$

Observe that the center coefficient of the filter, $w(0,0)$ aligns with the pixel at location (x,y).

General mask of size $m * n$:

Assume that

$$m = 2a+1$$
$$and$$
$$n = 2b+1$$

(where a, b are **positive integers**).

(Odd filters)

In general, linear spatial filtering of an image of size $M * N$ with a filter of size $m * n$ is given by the expression:

$$g(x,y) \;= \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t).\,f(x+s,y+t)$$

Where x and y are varied so that each pixel in w visits every pixel in f.

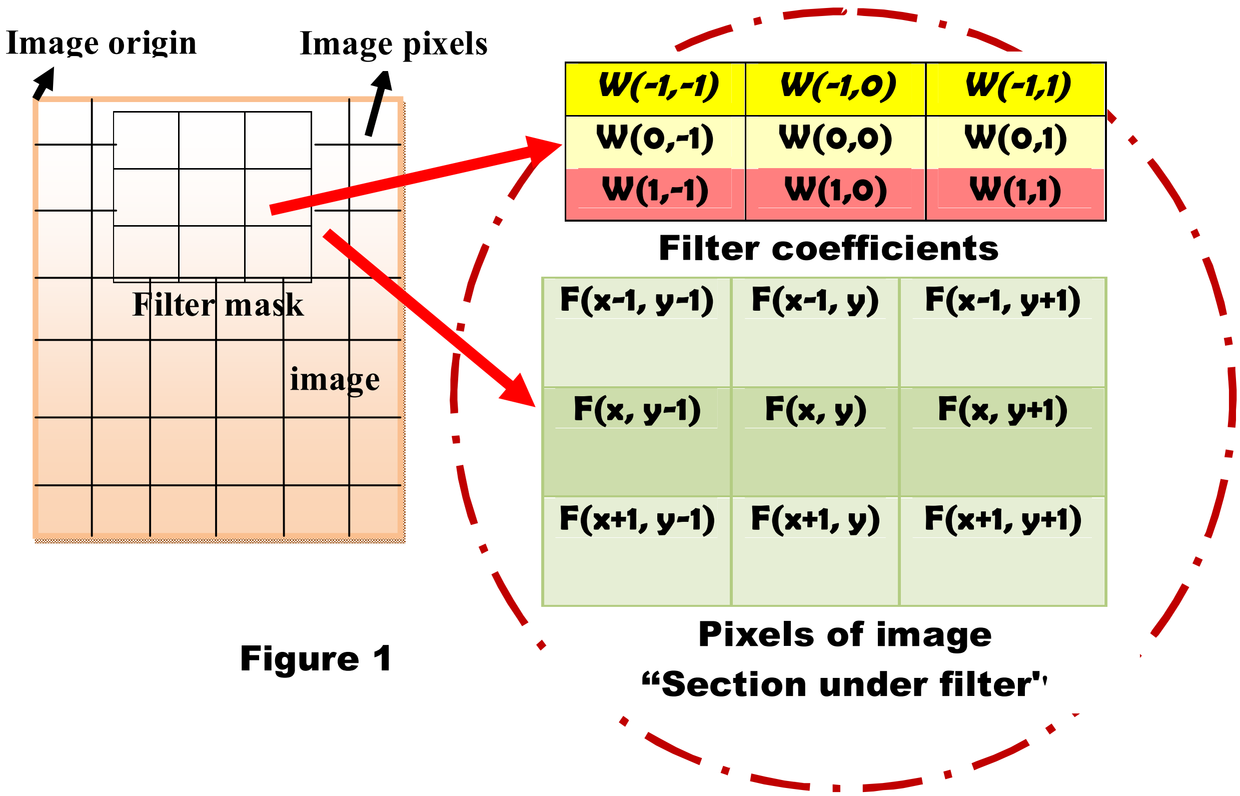
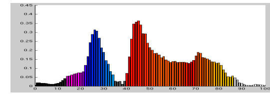


Figure 1

Spatial Correlation and convolution

- **Correlation**: the process of moving a filter mask over the image and computing the sum of products at each location.
- **Convolution**: the same process as correlation, except that the filter is first ***rotated by*** 180^0



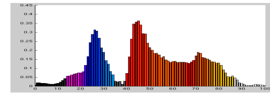
Example: 1-D illustration: (figure 2)

Assume that ***f*** is a 1-D function, and ***w*** is a filter

Correlation	Convolution
Origin f w	Origin f w rotated 180°
(a) 0 0 0 1 0 0 0 0 1 2 3 2 8	0 0 0 1 0 0 0 0 8 2 3 2 1 (i)
(b) 0 0 0 1 0 0 0 0	0 0 0 1 0 0 0 0 (j)
1 2 3 2 8	8 2 3 2 1
Starting position alignment	
Zero padding	
(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
1 2 3 2 8	8 2 3 2 1
(d) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
1 2 3 2 8	8 2 3 2 1
Position after one shift	
(e) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
1 2 3 2 8	8 2 3 2 1
Position after four shifts	
(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
1 2 3 2 8	8 2 3 2 1
Final position	
Full correlation result	Full convolution result
(g) 0 0 0 8 2 3 2 1 0 0 0 0	0 0 0 1 2 3 2 8 0 0 0 0 (o)
Cropped correlation result	Cropped convolution result
(h) 0 8 2 3 2 1 0 0	0 1 2 3 2 8 0 0 (p)

FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Figure 2



Notes:

- There are parts of the functions (images) that ***do not overlap*** (the solution of this problem is ***pad f*** with enough 0s on either side to allow each pixel in ***w*** to visit every pixel in ***f***.
- If the filter is of size ***m***, we need (***m-1***) 0s on either side of ***f***.
- The first value of correlation is the sum of products of ***f*** and ***w*** for the initial position (Figure 2.c).

  (The sum of product =0) this corresponds to a displacement ***x =0***
- To obtain the second value of correlation, we shift ***w*** are pixel location to the right (displacement ***x=1***) and compute the sum of products (result =0).
- The first nonzero is when ***x=3***, in this case the ***8*** in ***w*** overlaps the ***1*** in ***f*** and the result of correlation is ***8***.
- The full correlation result (figure 2.g) -12 values of ***x***
- To work with correlation arrays that are the same size as ***f***, in this case, we can crop the full correlation to the size of the original function. (Figure 2.h).
- The result of correlation is a copy of ***w***, but rotated by $\mathbf{180^0}$
- The correlation with a function with a discrete unit impulse yields a rotated version of the function at the location of the impulse.
- The convolution with a function with a discrete unit impulse yields a copy of that function at the location of the impulse.

Correlation and convolution with images

- With a filter of size ***m*n***, we pad the image with a minimum of ***m-1*** rows of 0s at the top and the bottom, and ***n-1*** columns of 0s on the left and right.
- If the filter mask is ***symmetric,*** correlation and convolution yield the same result.

Summary:

- **Correlation** of a filter $w(x, y)$ of size $m * n$ with an image $f(x, y)$ denoted as

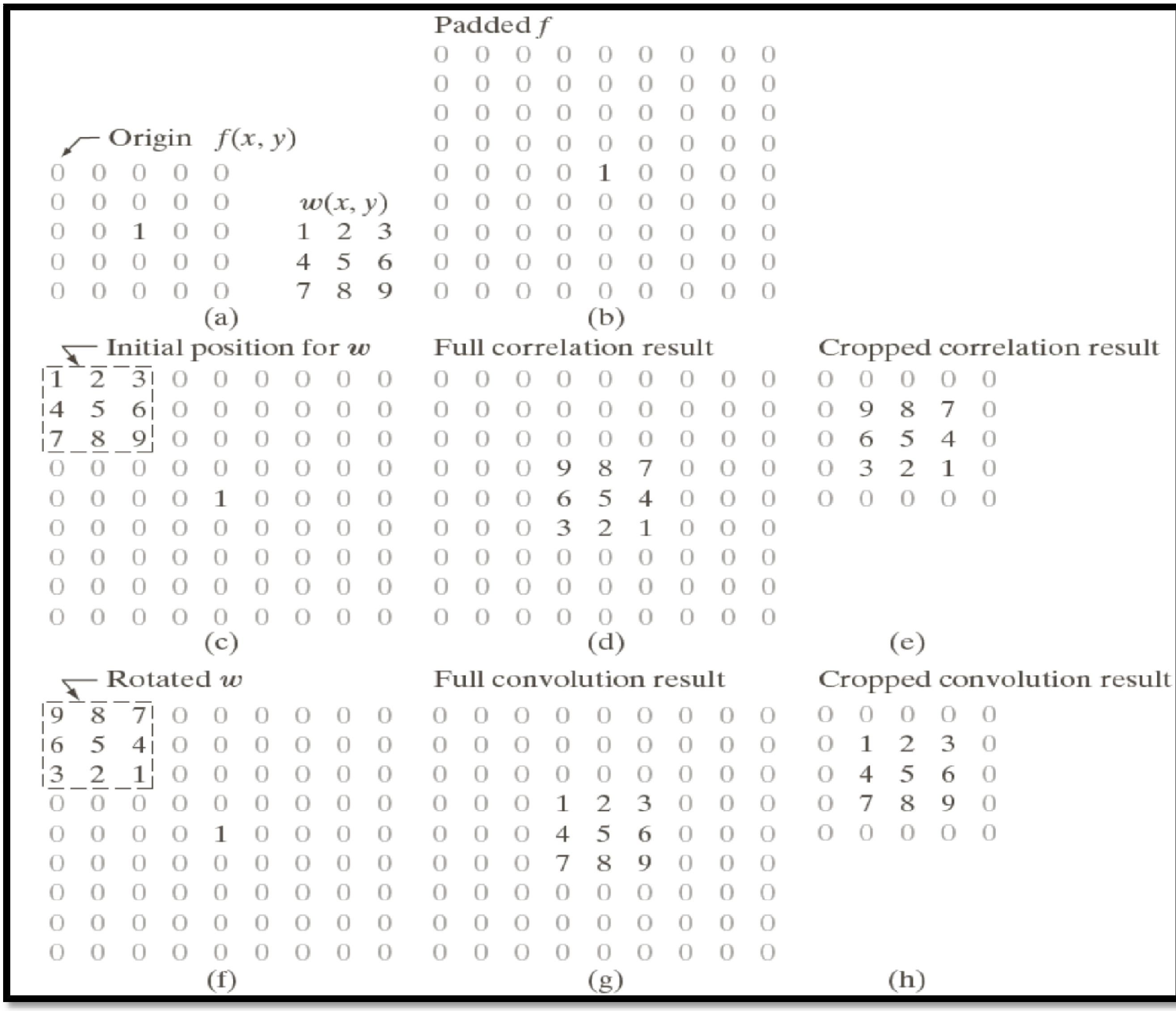
$$W(x,y)\,{}^{\circ}f(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

- In similar manner, the **convolution** of $w(x, y)$ and $f(x, y)$ denoted by $w(x, y) * f(x, y)$ is given by:

$$W(x,y) * f(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x-s,y-t)$$

Where the minus sign on the right flip (rotate by $\mathbf{180^0}$)

(We can flip and shift either f or w)



Vector Representation of linear filtering:

- **Correlation**

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_{mn} z_{mn} = \sum_{k=1}^{mn} w_k z_k = \boldsymbol{w^T}\,\mathrm{Z}$$

- ✓ R- the response of a mask
- ✓ W_k– the coefficients of an m ∗ n filter
- ✓ Z_k – the corresponding image intensities encompassed by the filter

- **Convolution**

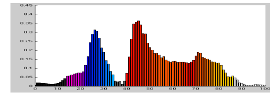
We simply rotate the mask by $\mathbf{180^0}$

Example: The general 3*3 mask equation:

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{k=1}^{9} w_k z_k = \boldsymbol{w^T}\,\mathrm{Z}$$

Where:

W and Z are g-dimensional vectors (mask and image)



Another representation of 3*3 filter mask

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Example of filters masks

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.