Linear Systems: Discrete case & 2D

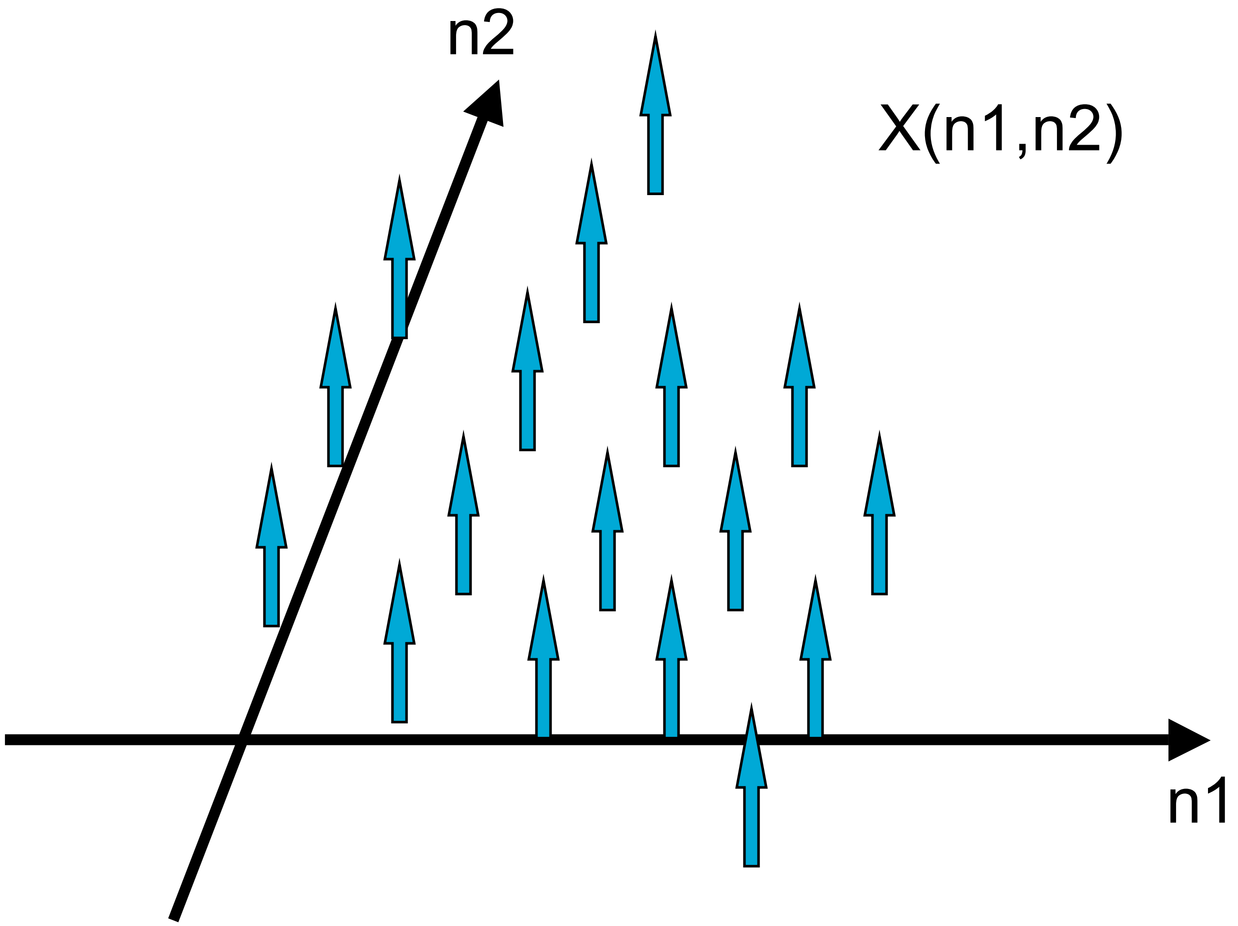
Linear systems-review

Part 1: Review from G&W (continuous case)

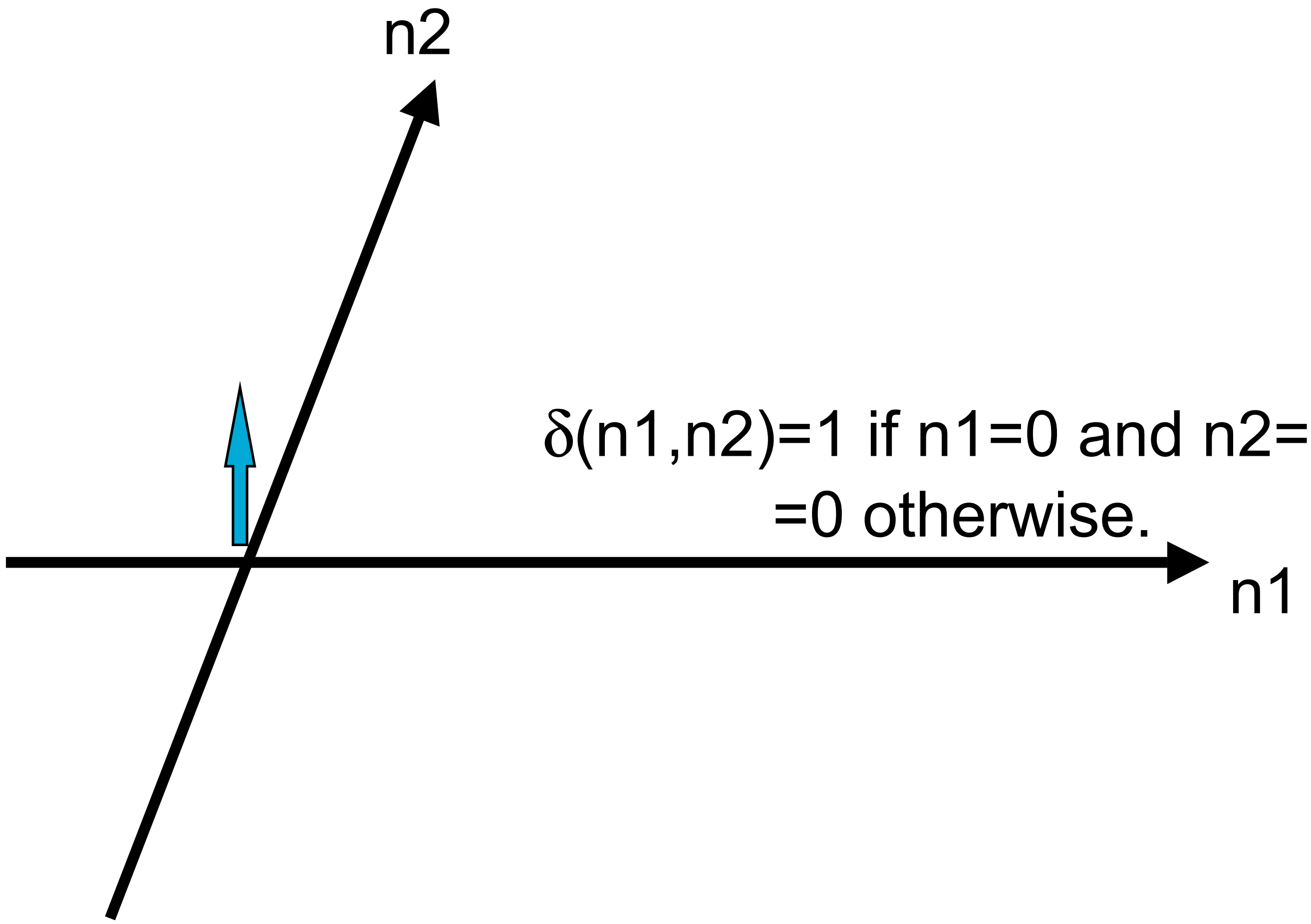
Part 2: Discrete case & 2D

- 2D impulse function
- Line function
- Step function
- Linear systems and Shift invariance
- Impulse Response of LSI Systems
- 2-D Convolution

2-D Systems

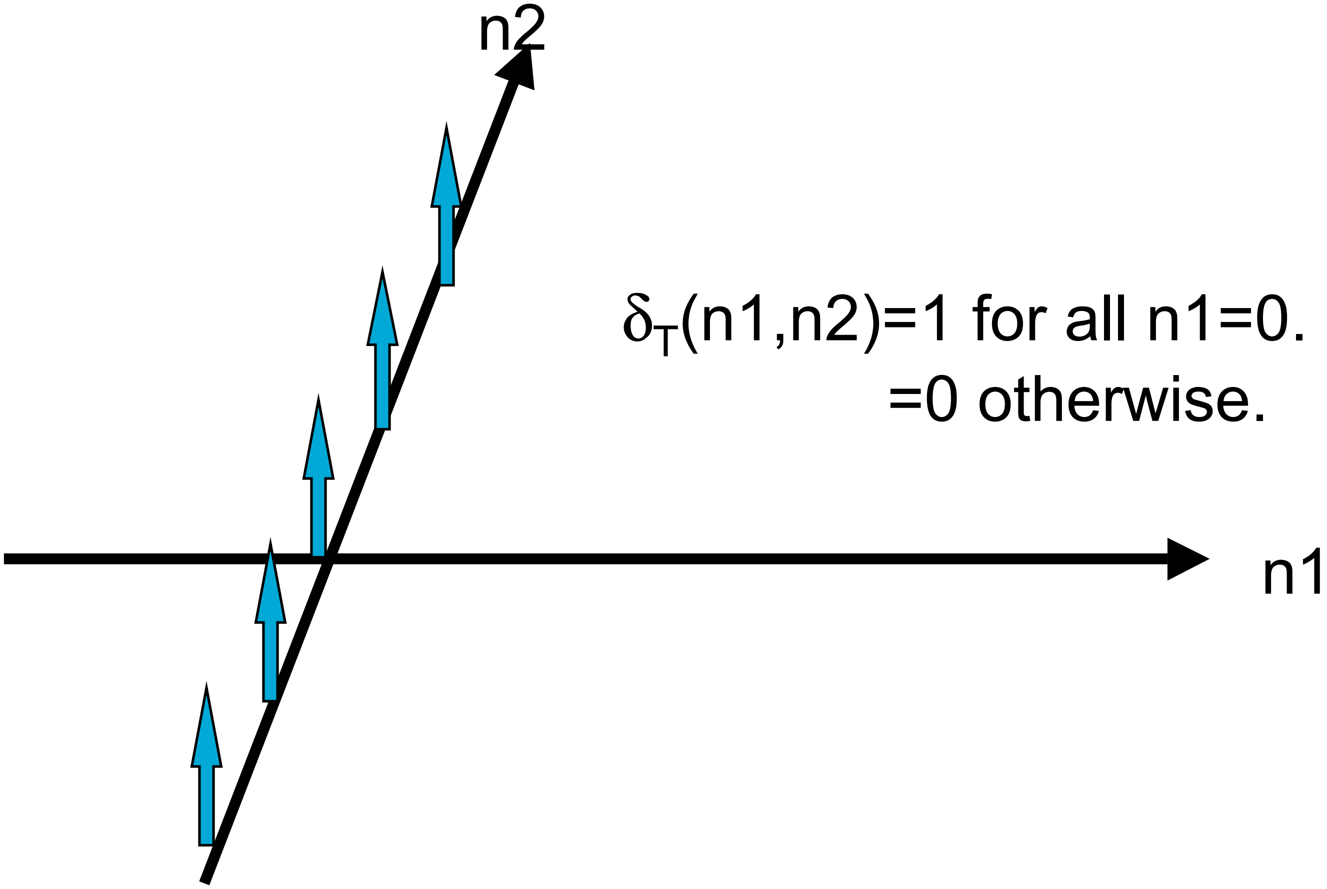


Impulse Function(Kronecker Delta)



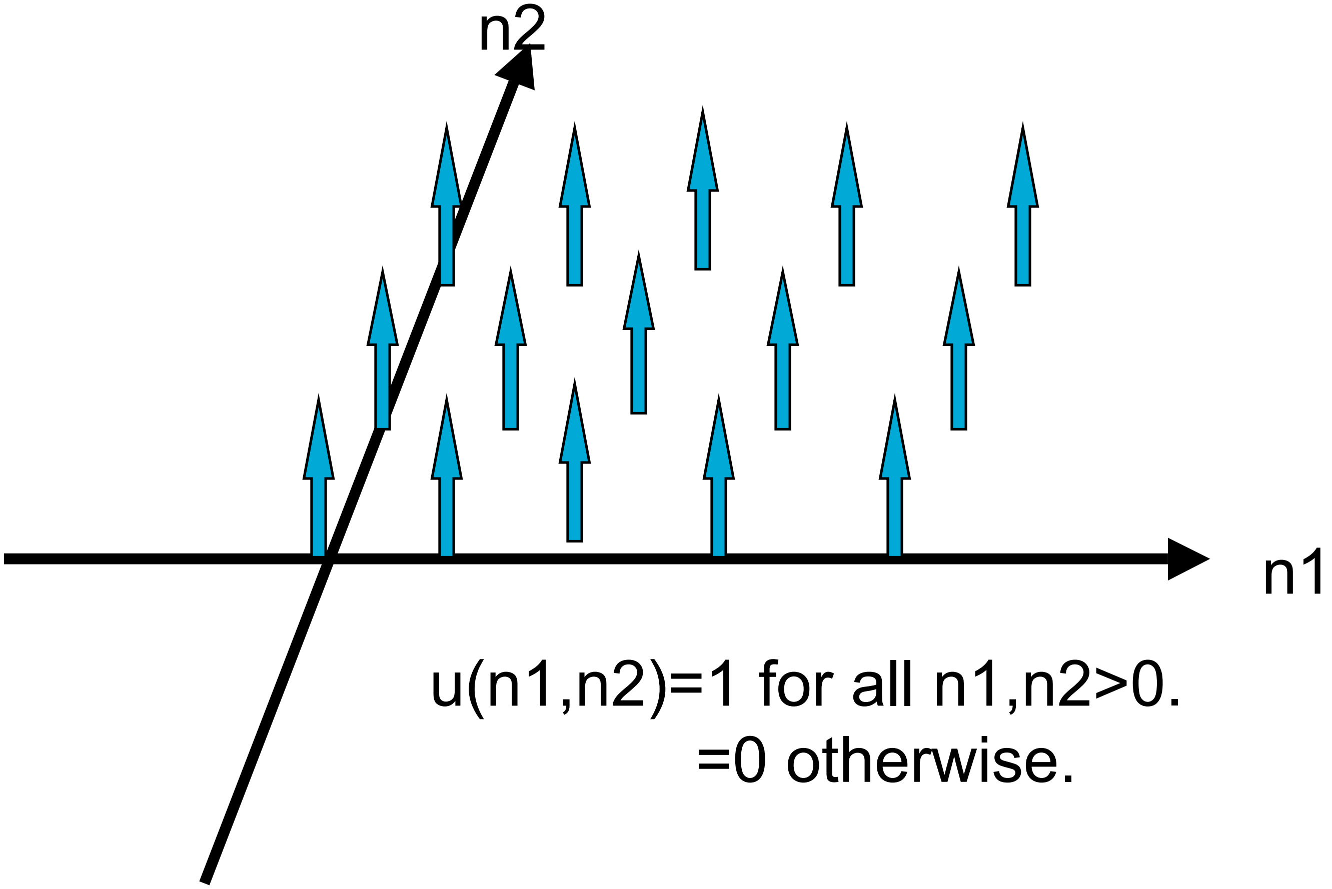
$\delta(n1,n2)=1$ if $n1=0$ and $n2=0$.
$=0$ otherwise.

Line Impulse



$\delta_T(n1,n2)=1$ for all $n1=0$.
$=0$ otherwise.

Unit Step Function



u(n1,n2)=1 for all n1,n2>0.
=0 otherwise.

“ System”

An input-output relationship is called a system if there is a unique output for any given input.

$$Y(n1,n2) = T[X(n1,n2)]$$

Linear Systems

The linearity of a system T is defined as

Linearity:

$T[a\ X_1(n1,n2) + b\ X_2(n1,n2)] = a\ Y_1 + b\ Y_2$

(i.e., principle of superposition holds).

Are these linear?

(a) $y(m,n) = x(m,n)\ g(m,n)$

(b) $y(m,n) = [x(m,n)]^2$

Linear Shift Invariant Systems

Shift Invariance:

$T[X(m-k, n-l)] = Y(m-k, n-l)$ where

$Y(m,n) = T[X(m,n)]$.

A LSI system is completely characterized by its response to the impulse function $\delta(m,n)$.

Convolution

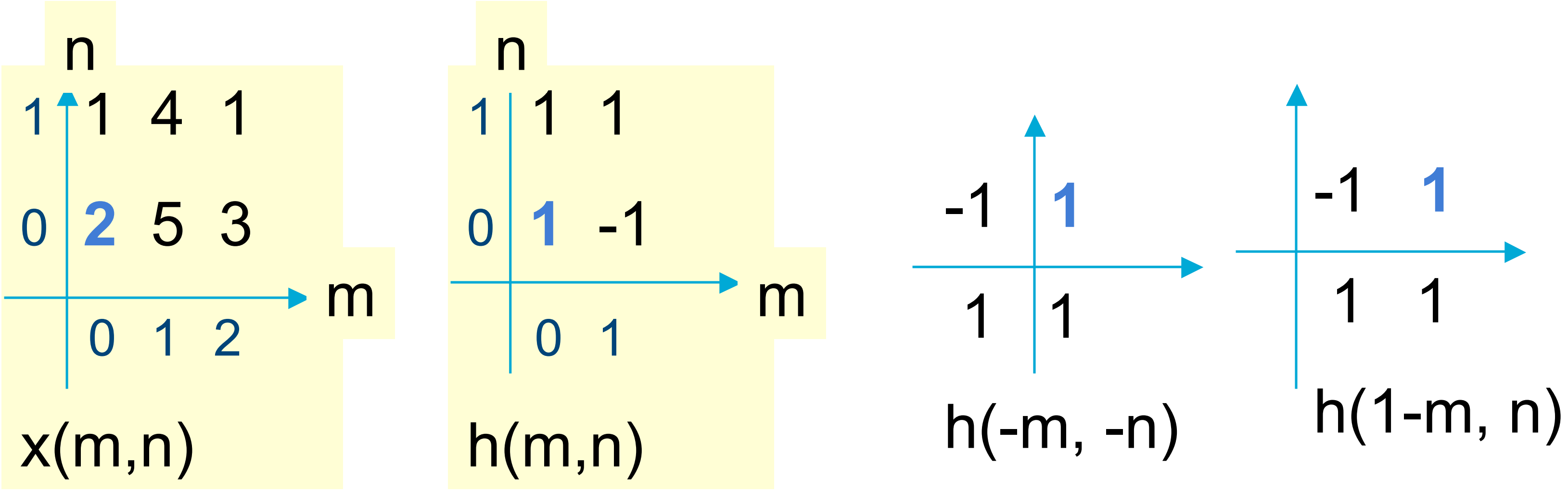
Let $h(n_1,n_2) = T[\delta(n_1,n_2)]$; $y(n_1,n_2) = T[x(n_1,n_2)]$; then

$h(n_1 - k_1, n_2 - k_2) = T[\ \delta(n_1 - k_1, n_2 - k_2)\ ]$, and

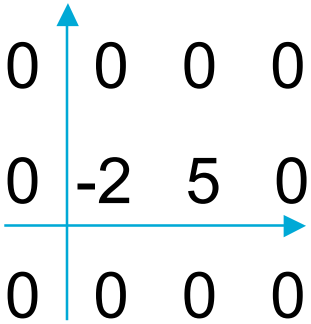
$$\begin{aligned} y(n_1,n_2) &= T\left[\sum_{k_1=-\infty}^{\infty}\sum_{k_1=-\infty}^{\infty} x(k_1,k_2)\delta(n_1-k_1,n_2-k_2)\right] && (1) \\ &= \sum_{k_1=-\infty}^{\infty}\sum_{k_1=-\infty}^{\infty} x(k_1,k_2)T[\delta(n_1-k_1,n_2-k_2)] && (2) \\ &= \sum_{k_1=-\infty}^{\infty}\sum_{k_1=-\infty}^{\infty} x(k_1,k_2)h(n_1-k_1,n_2-k_2) && (3) \end{aligned}$$

$$y(n_1,n_2) = h(n_1,n_2) * x(n_1,n_2)$$

Convolution: example

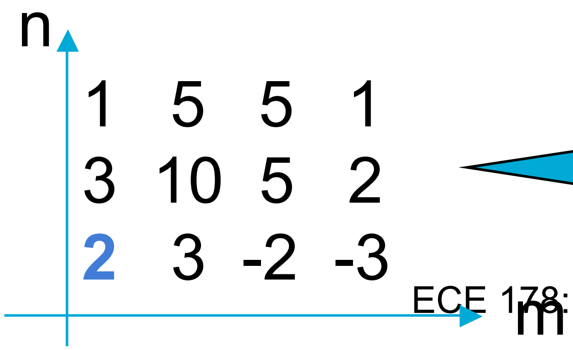


$y(1,0) = \Sigma_{k,l}\, x(k,l)h(1-k, -l) =$



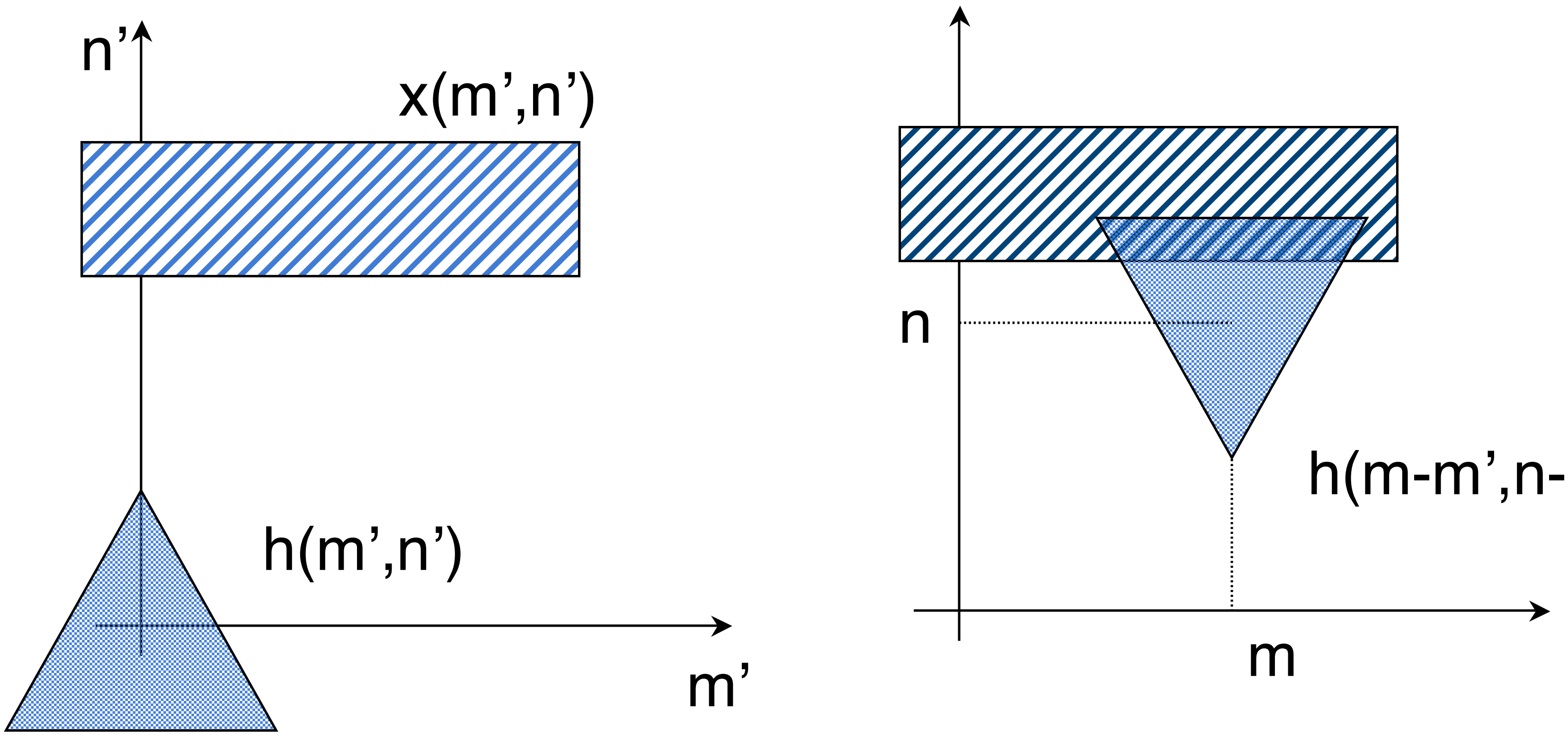
= 3

y(m,n)=



verify!

Discrete Convolution in 2D



output=sum of the product
of the two in the overlapped region.

Correlation vs Convolution

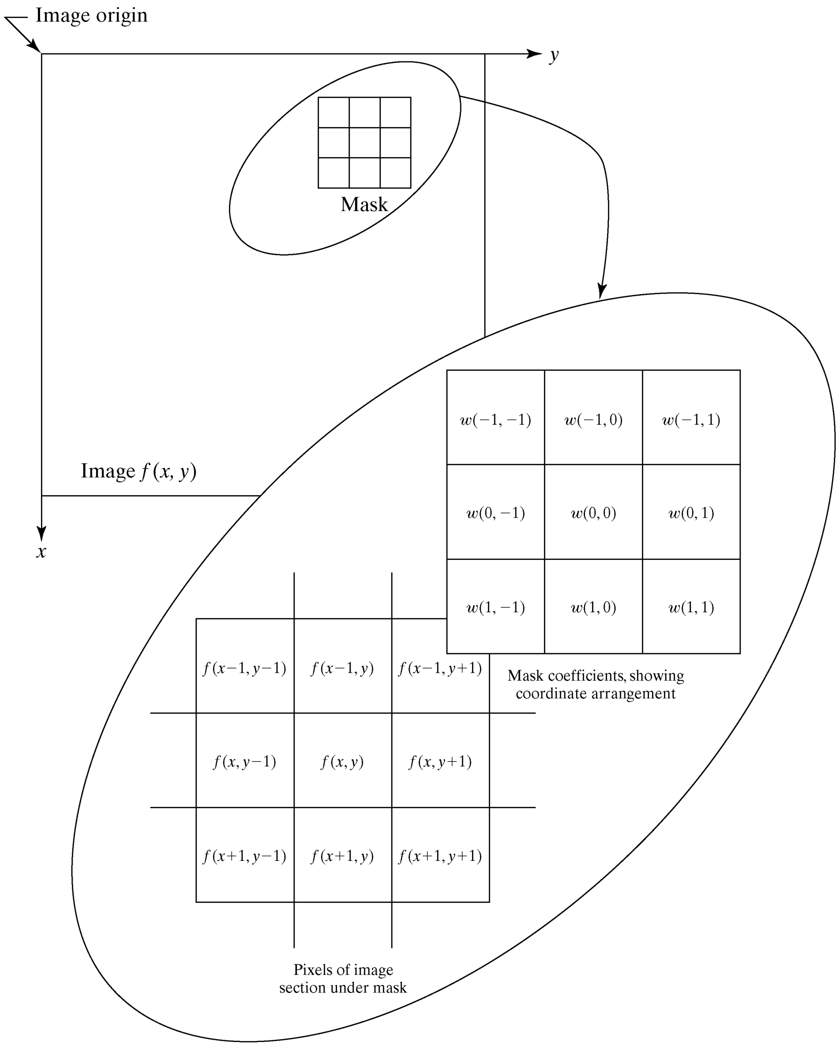
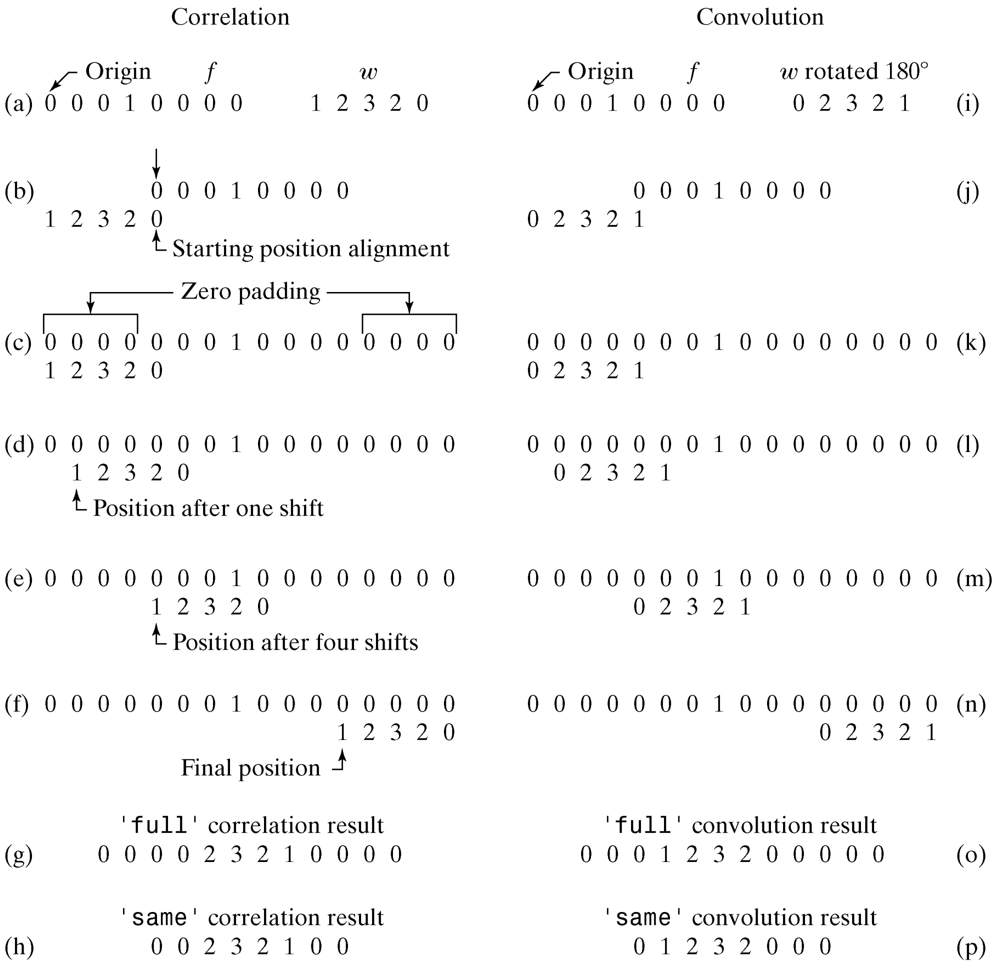
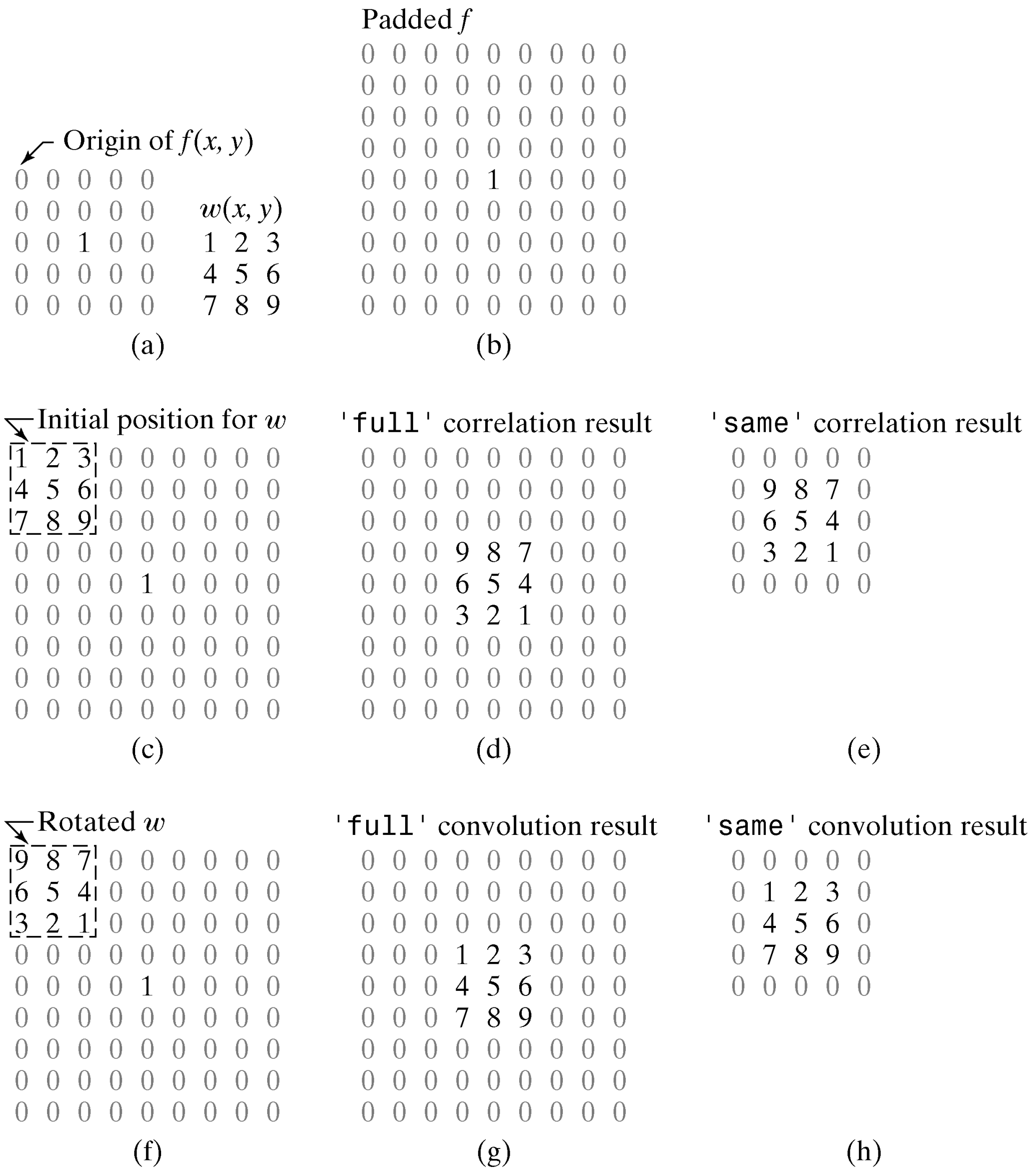


FIGURE 3.12 The mechanics of linear spatial filtering. The magnified drawing shows a 3 × 3 mask and the corresponding image neighborhood directly under it. The neighborhood is shown displaced out from under the mask for ease of readability.

1-D example


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

2-D example



Origin of $f(x, y)$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(b)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

`'full'` correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

`'same'` correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(f)

`'full'` convolution result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

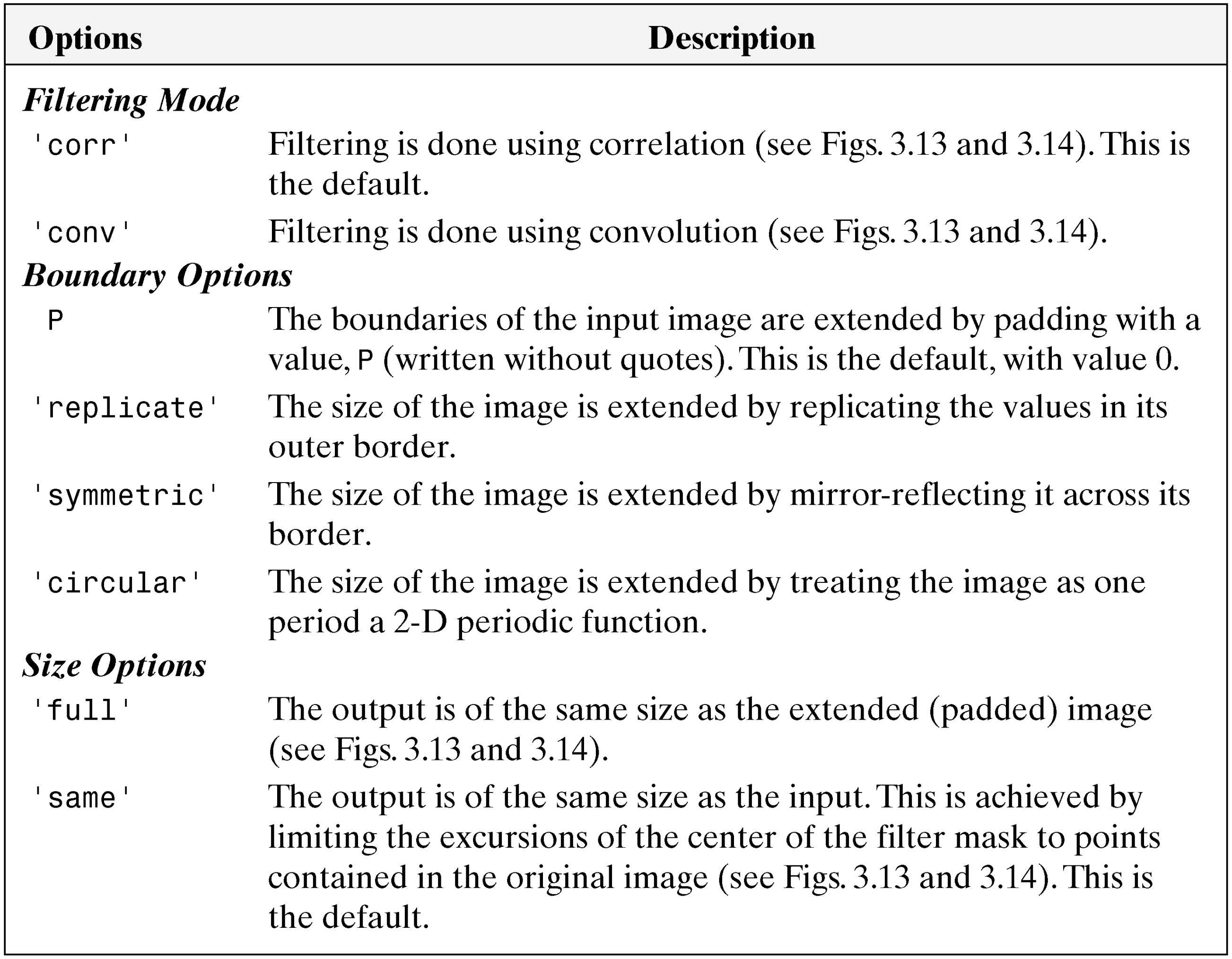
(g)

`'same'` convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

(h)

Matlab command: *imfilter*



Options	Description
Filtering Mode	
`'corr'`	Filtering is done using correlation (see Figs. 3.13 and 3.14). This is the default.
`'conv'`	Filtering is done using convolution (see Figs. 3.13 and 3.14).
Boundary Options	
`P`	The boundaries of the input image are extended by padding with a value, `P` (written without quotes). This is the default, with value 0.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'circular'`	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
`'full'`	The output is of the same size as the extended (padded) image (see Figs. 3.13 and 3.14).
`'same'`	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 3.13 and 3.14). This is the default.