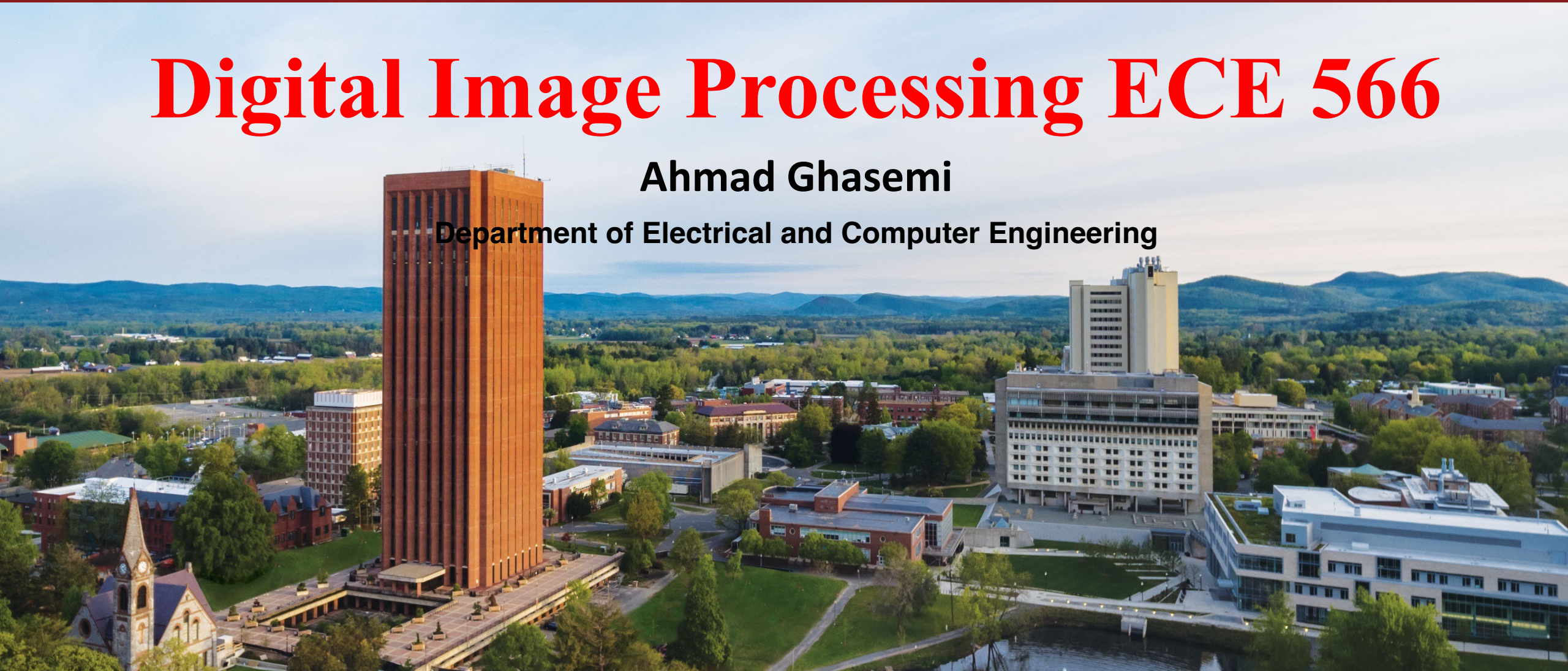
University of Massachusetts Amherst

Digital Image Processing ECE 566

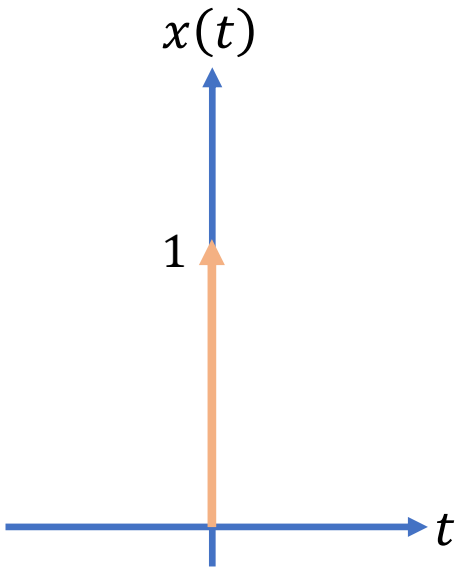
Ahmad Ghasemi

Department of Electrical and Computer Engineering



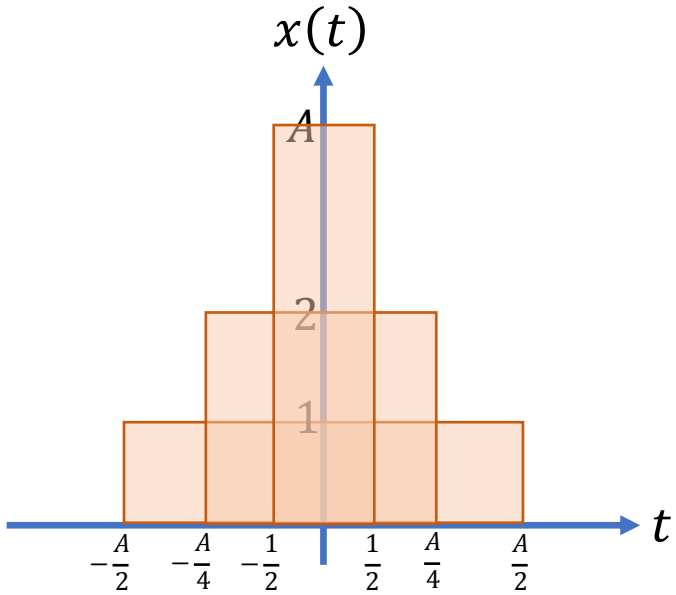
Impulse Function $\delta(t)$

- ✓ AKA a Dirac delta function.
- ✓ It is a pulse having a total area of 1 and its amplitude goes to infinity.



Impulse function at $t = 0$

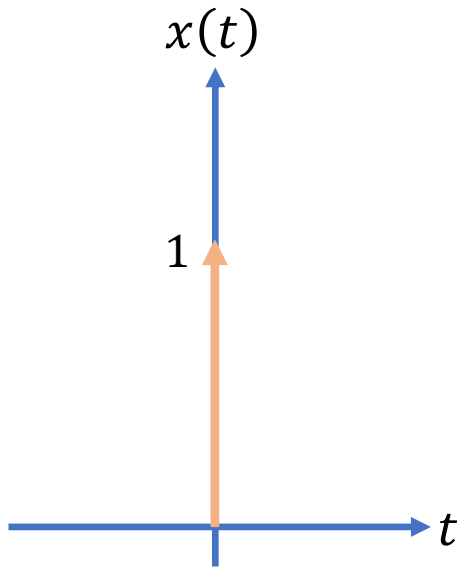
denoted as $\delta(t)$



Rectangular pulses with the same area A

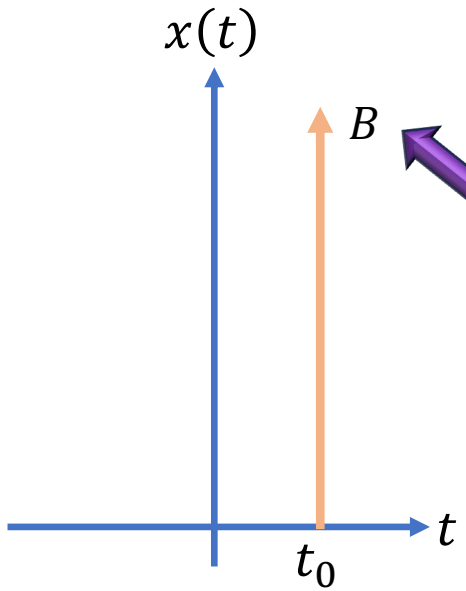
Impulse Function $\delta(t)$

- ✓ AKA a Dirac delta function.
- ✓ It is a pulse having a total area of 1 and its amplitude goes to infinity.



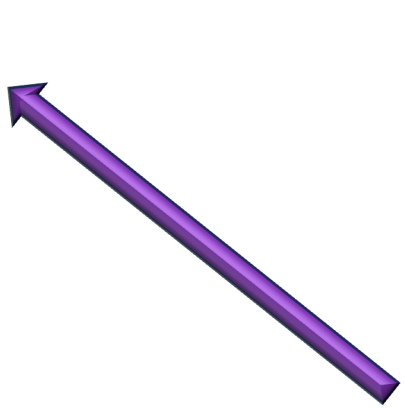
Impulse function at $t = 0$

denoted as $\delta(t)$

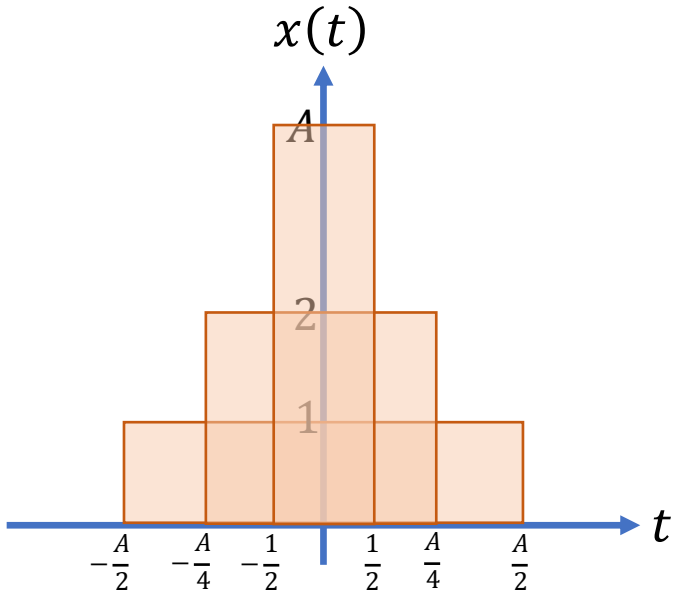


Impulse function at $t = t_0$

denoted as $\mathrm{B}\delta(t - t_0)$



Area under the impulse



Rectangular pulses with the same area A

Impulse Function $\delta(t)$: Properties

✓ Area

$$\int_{t=-\infty}^{\infty} \mathrm{B}\delta(t-t_0)dt = B \int_{t=-\infty}^{\infty} \delta(t-t_0)dt = B$$

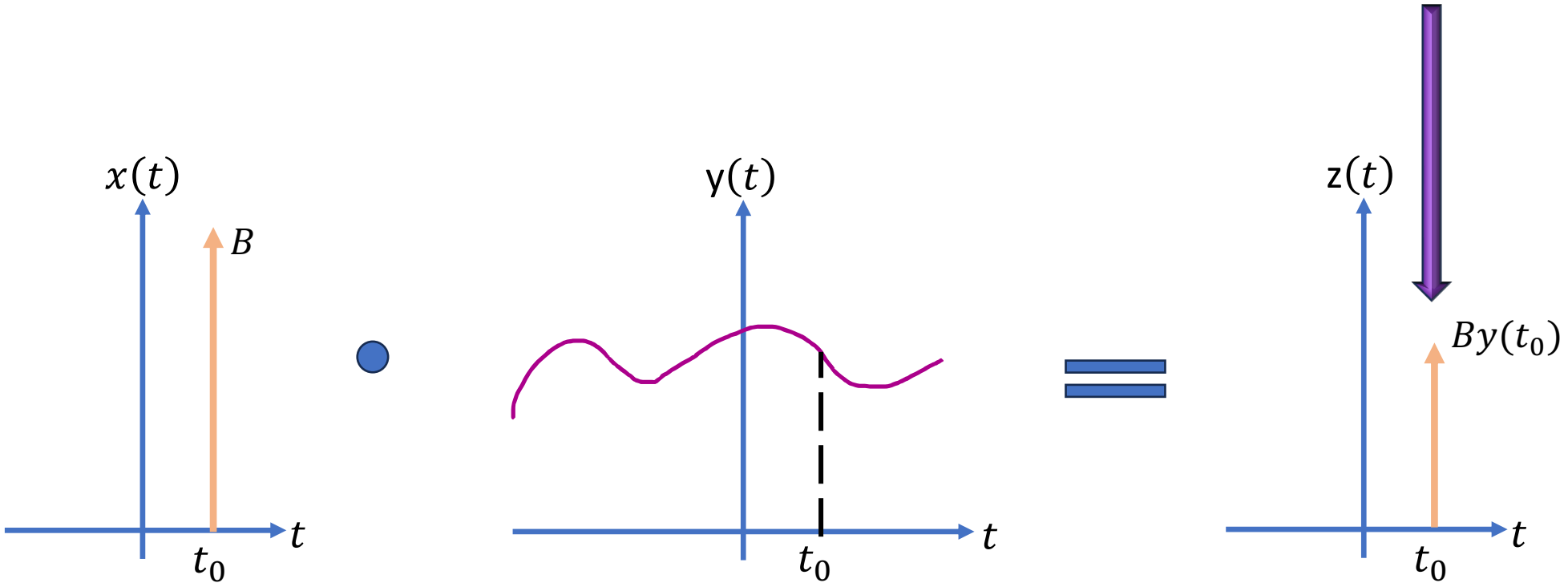
✓ Fourier Transform

$$\mathrm{B}\delta(t-t_0) \Leftrightarrow Be^{-i\omega t_0} \quad \text{where} \quad \omega = 2\pi f$$

Why?

Impulse Function $\delta(t)$: Sampling

- ✓ Multiplication of an impulse and a continuous function leads to scaling of the original impulse
- ✓ The scale factor corresponds to the sample value of the continuous function at the impulse location

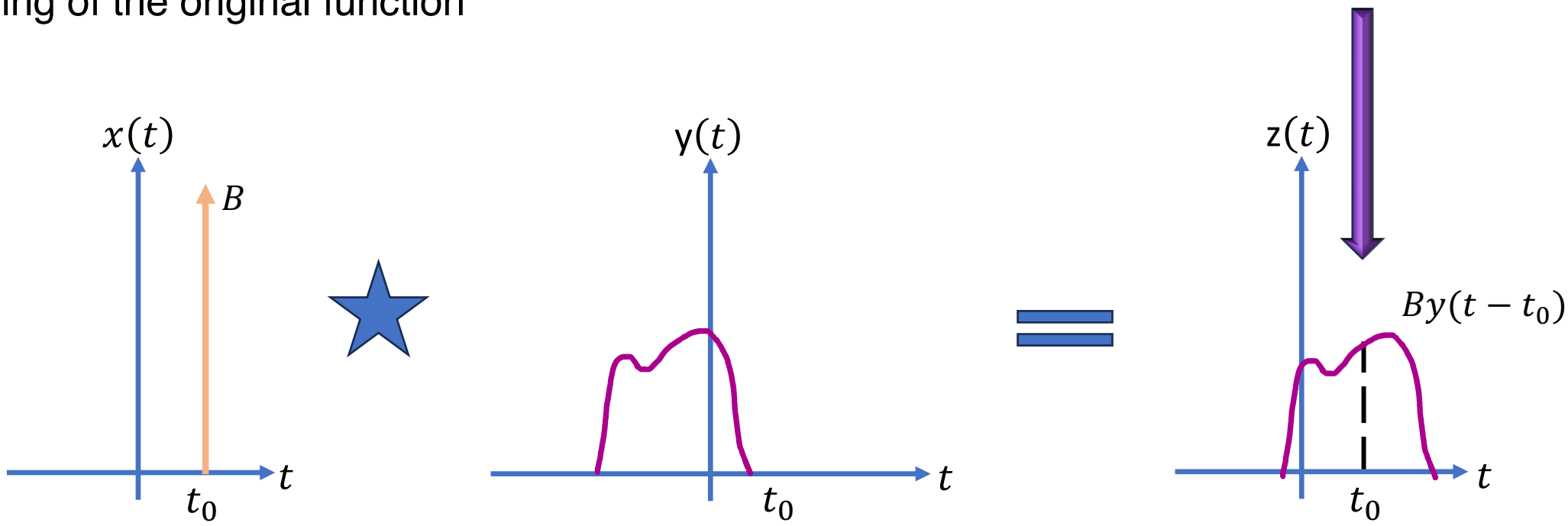


$$\mathrm{B}\delta(t-t_0)y(t) = \mathrm{B}y(t_0)\delta(t-t_0)$$

University of Massachusetts Amherst

Impulse Function $\delta(t)$: Convolution

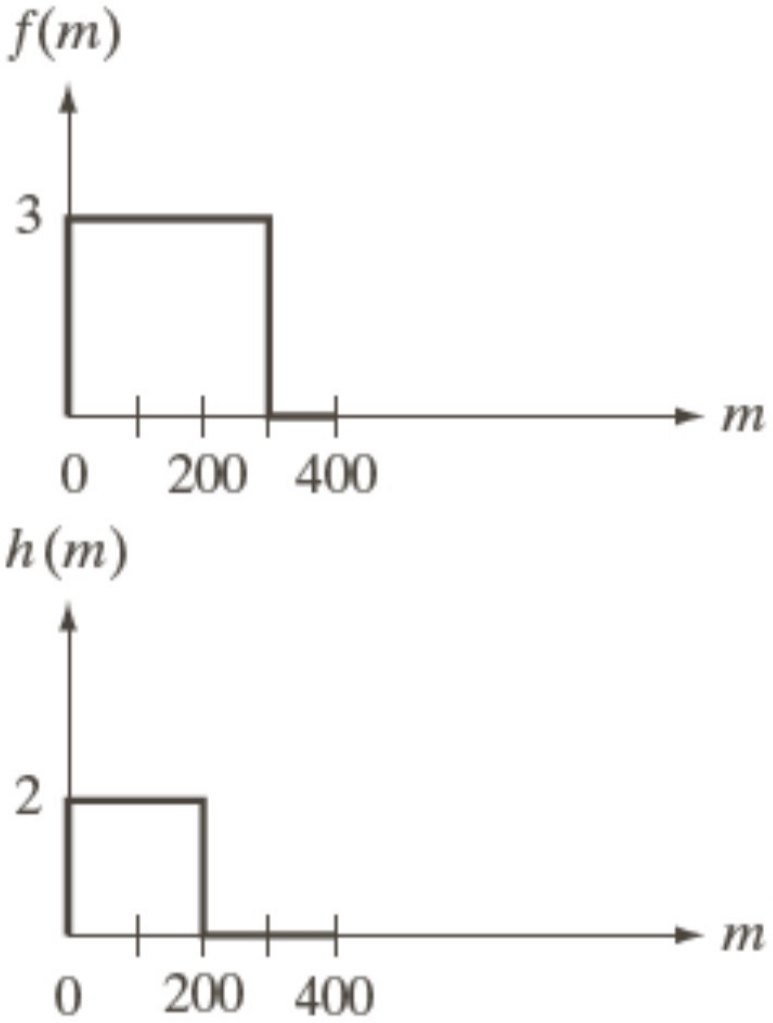
- ✓ Convolution of an impulse and a function leads to shifting and scaling of the original function
- ✓ The scale factor corresponds to the area of the impulse
- ✓ The shift value corresponds to the location of the impulse



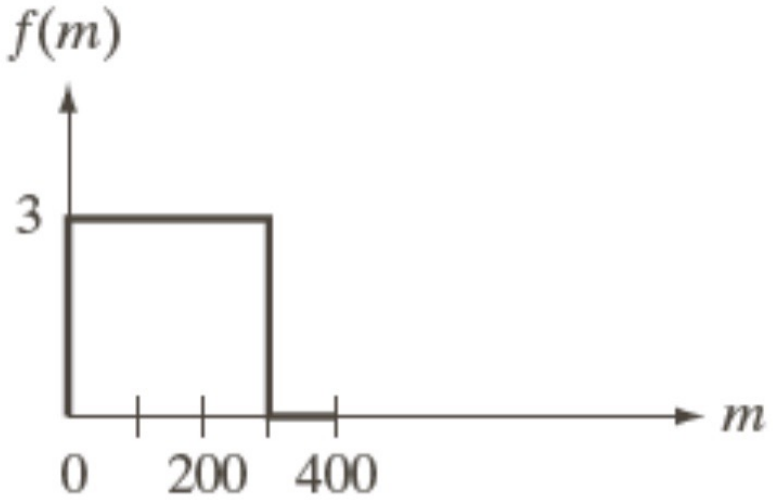
$$f(t) \bigstar g(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \Rightarrow \mathrm{B}\delta(t-t_0) \bigstar y(t) = \mathrm{B}y(t-t_0)$$

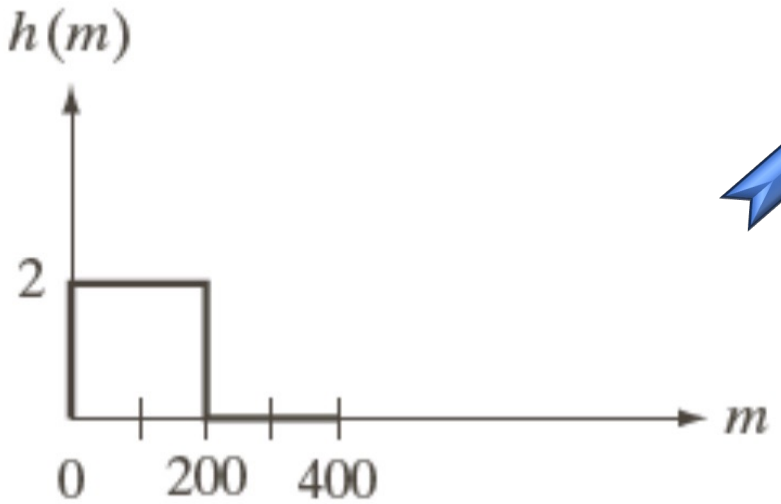
University of Massachusetts Amherst

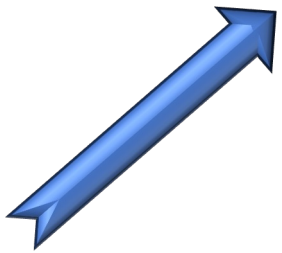
Convolution

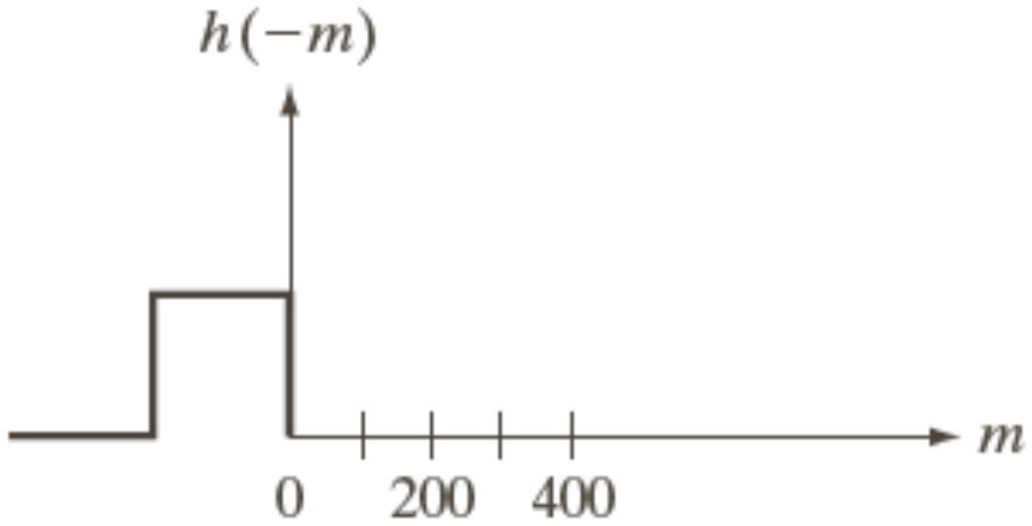


Convolution

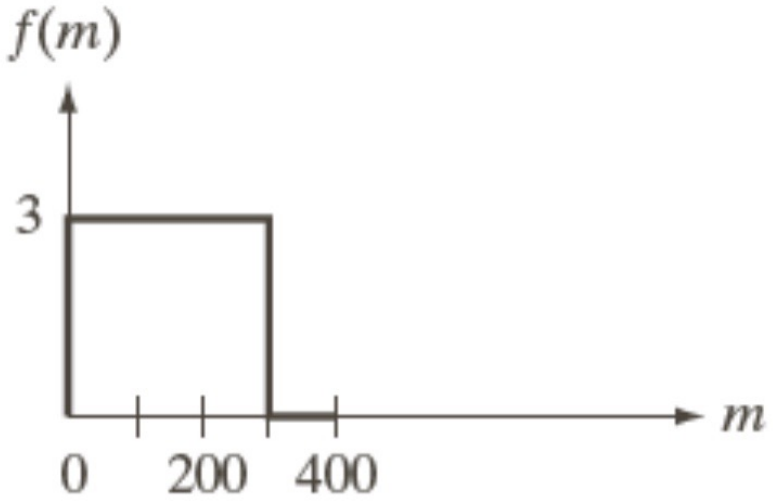


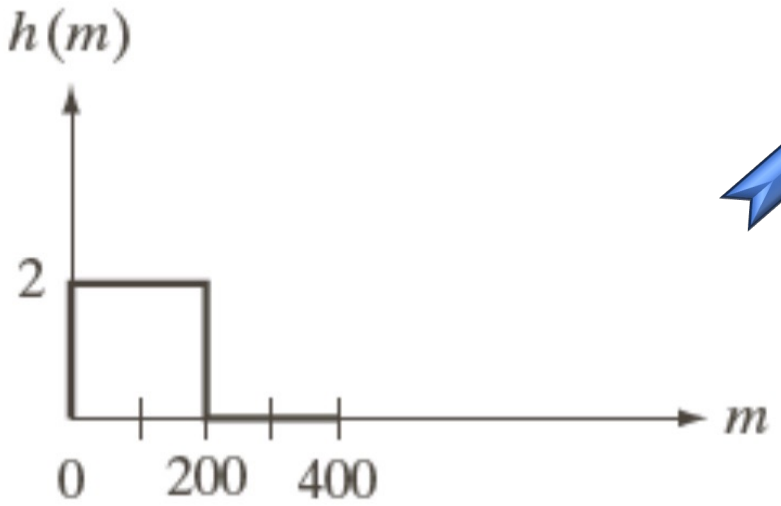


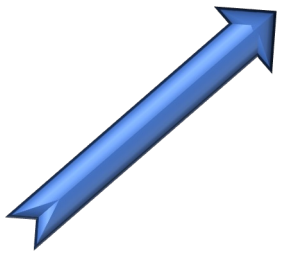


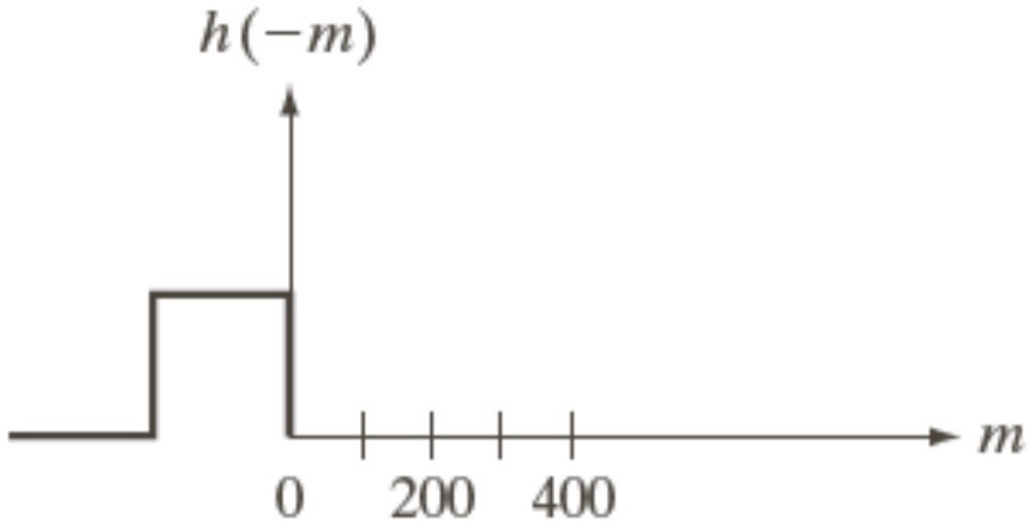


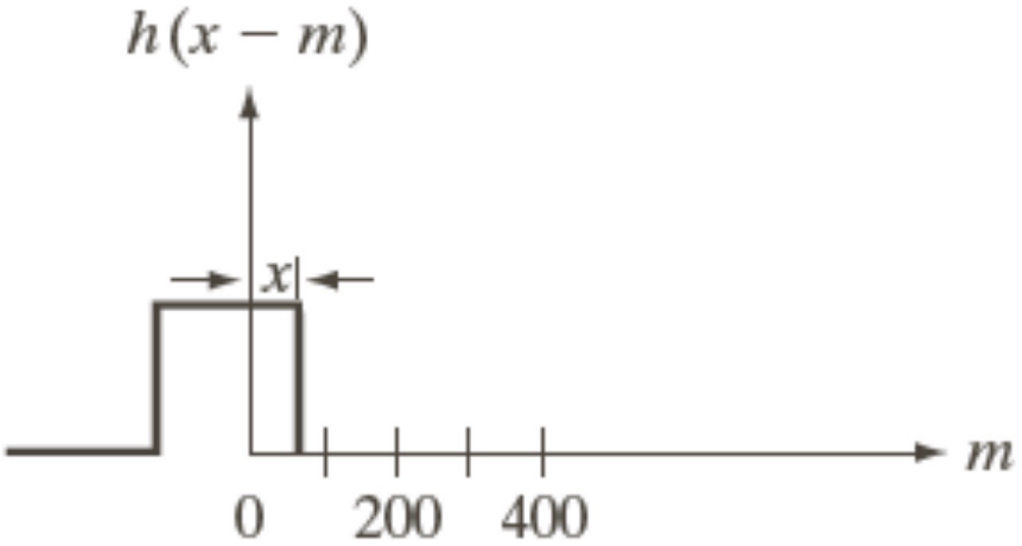
Convolution



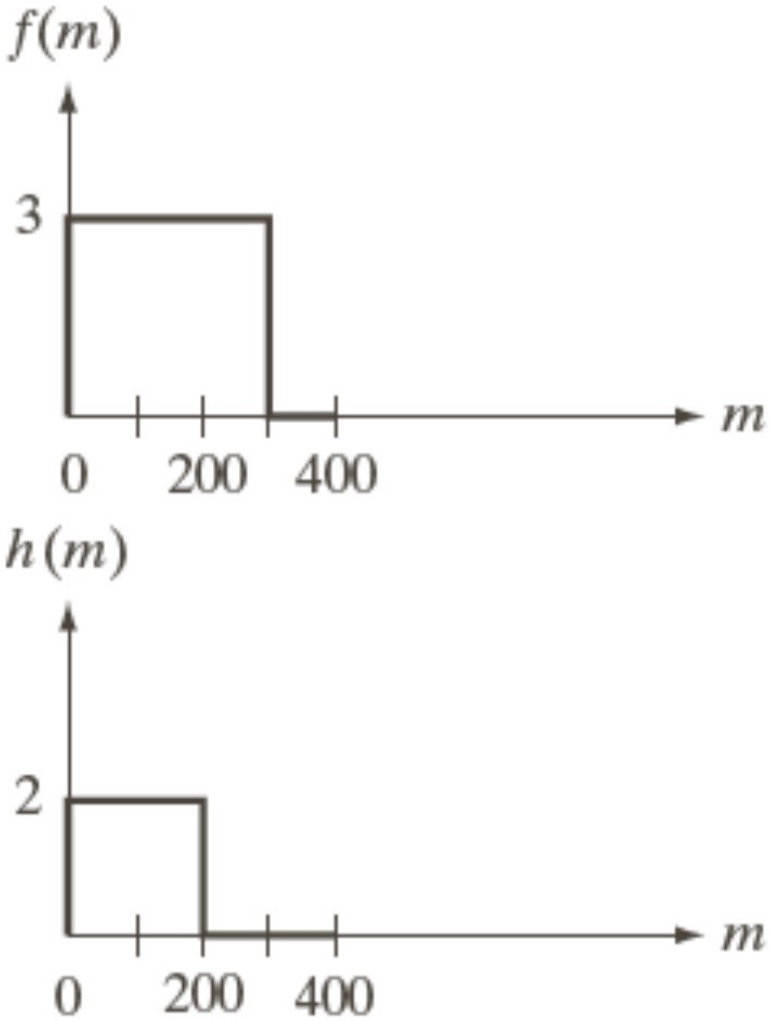


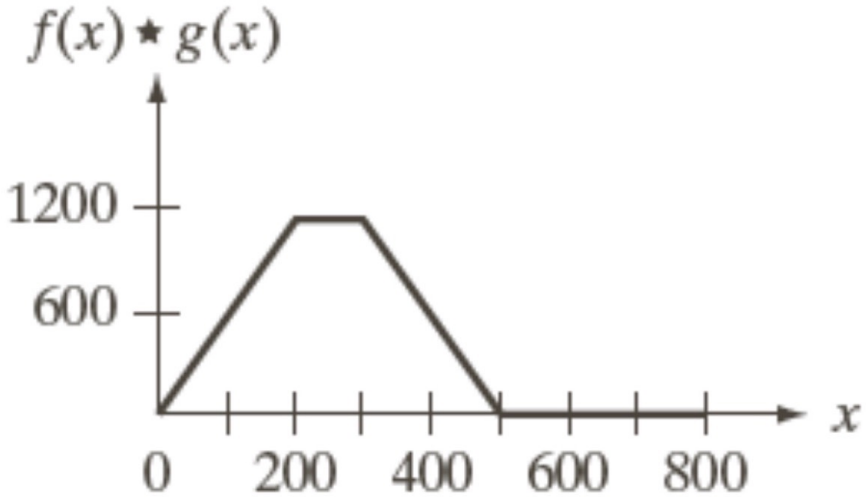




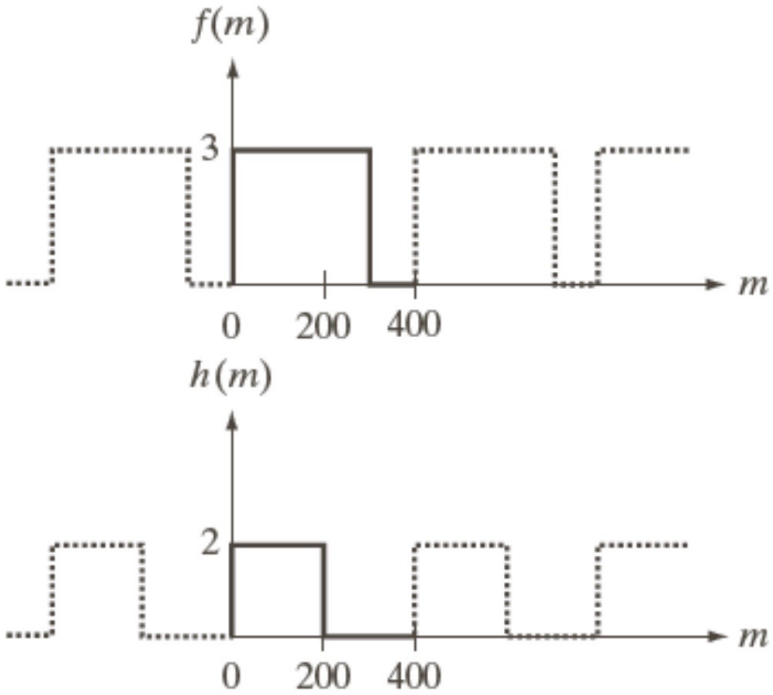


Convolution



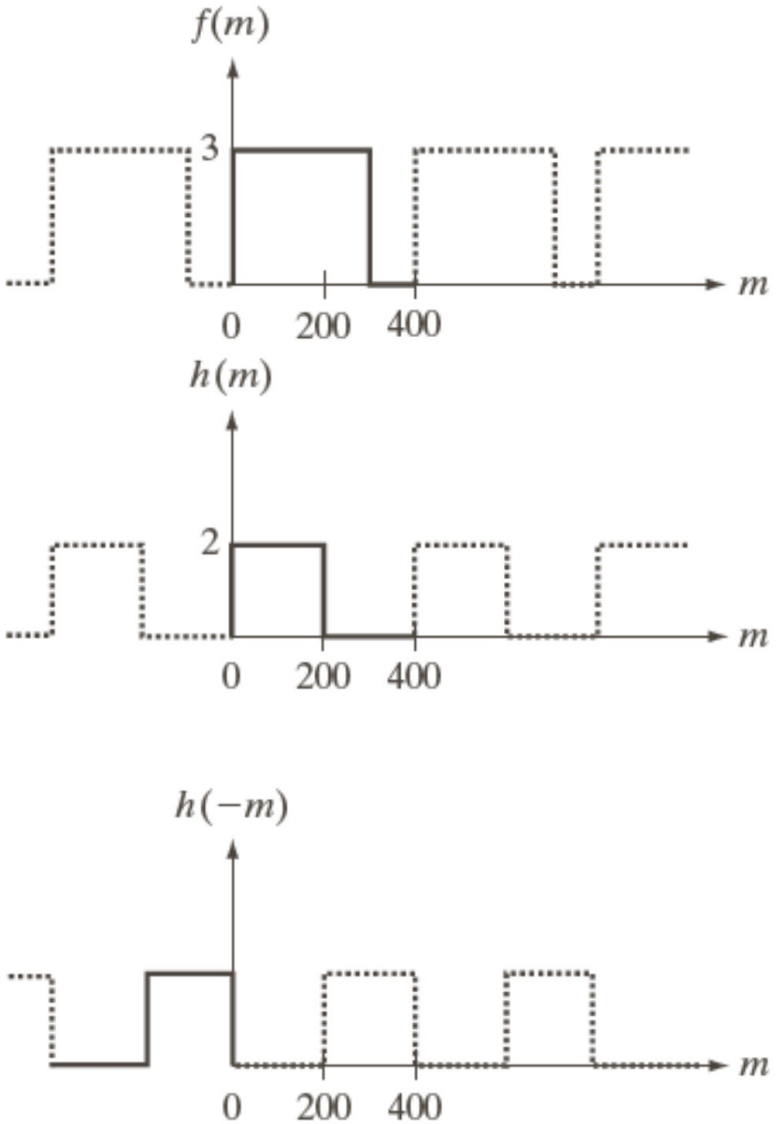


Circular Convolution



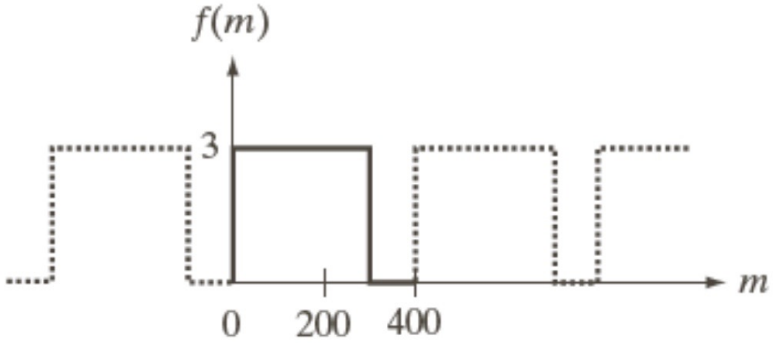
University of Massachusetts Amherst

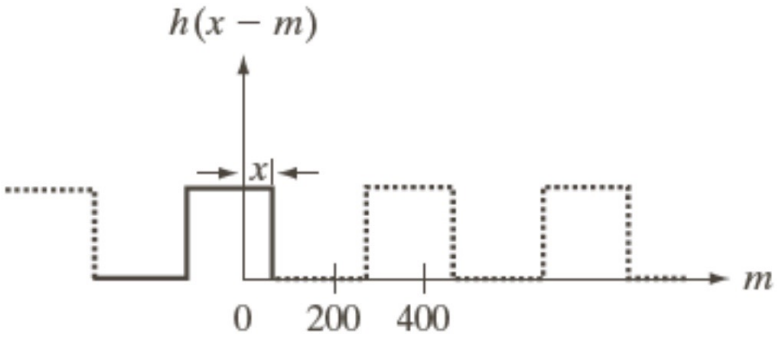
Circular Convolution

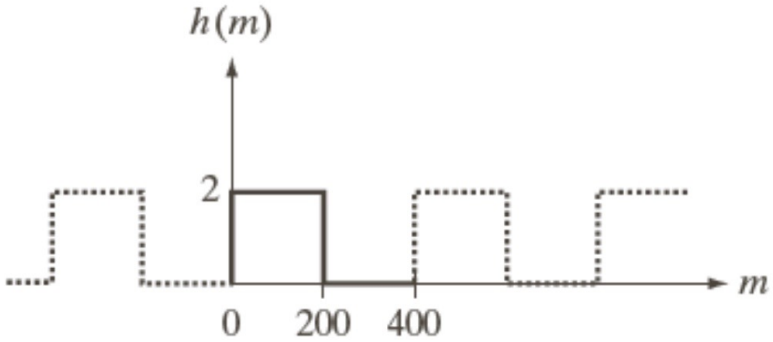


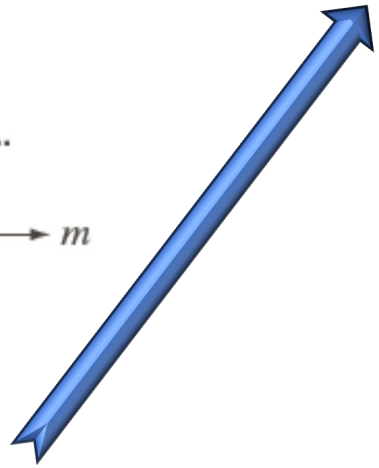
University of Massachusetts Amherst

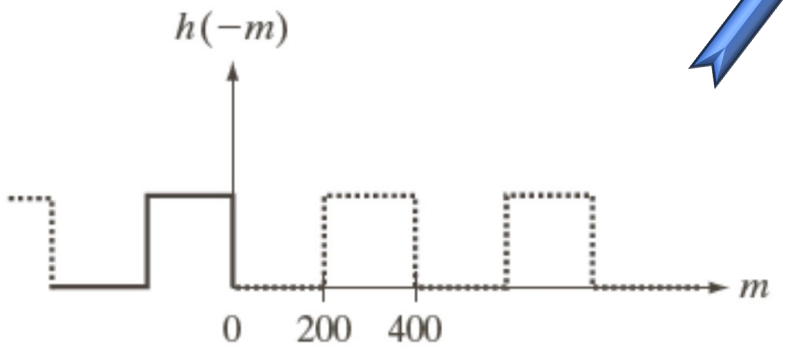
Circular Convolution



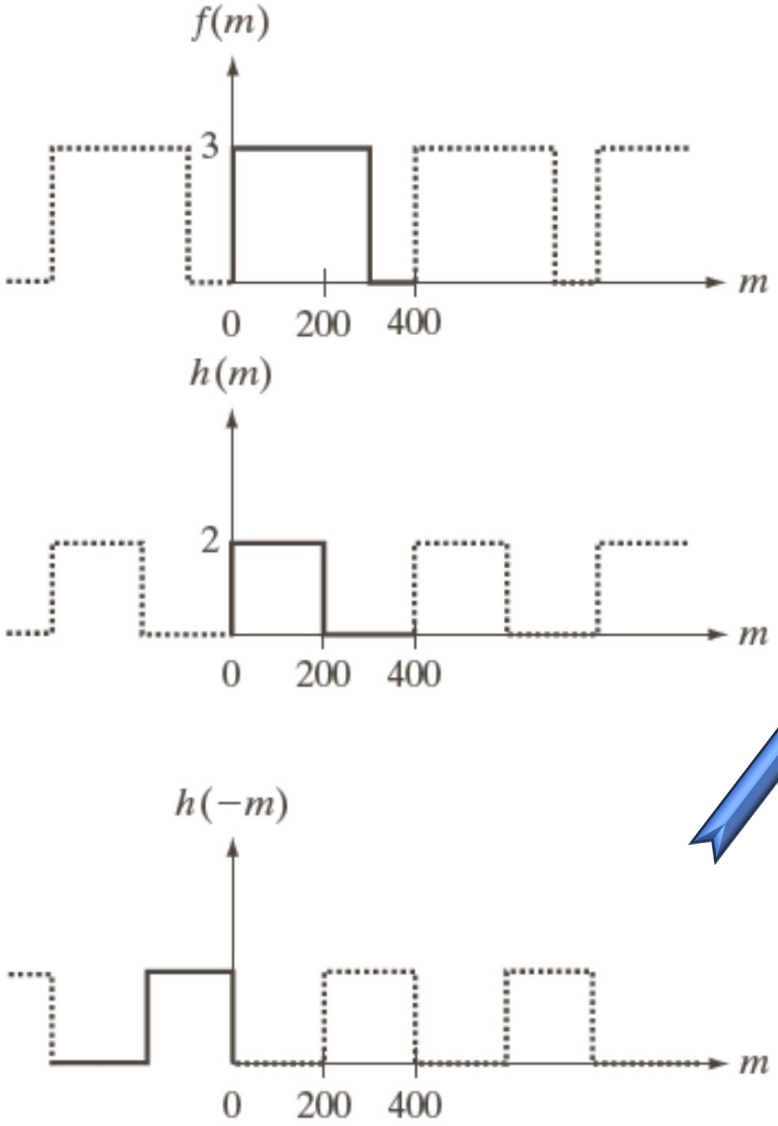


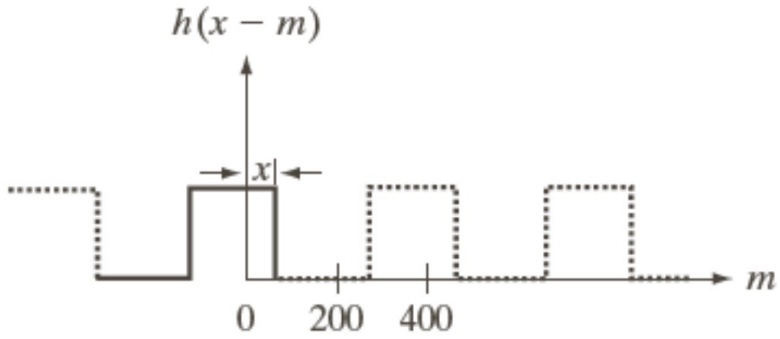


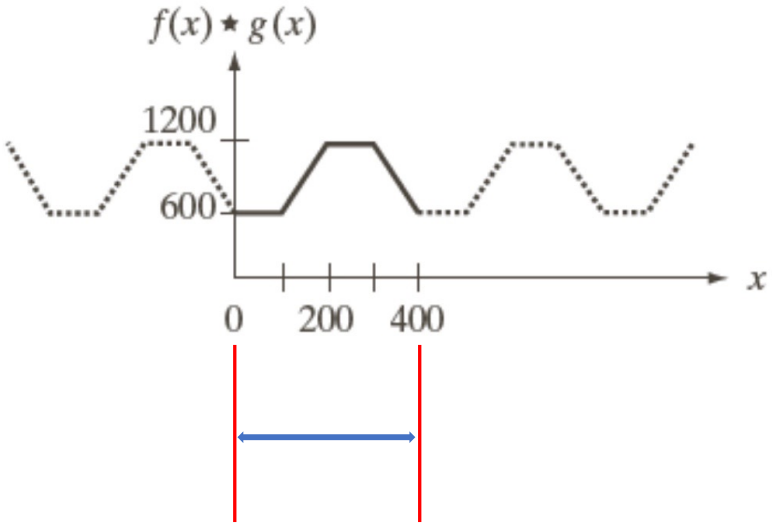




Circular Convolution







Range of FT computation

2D discrete convolution theorem

$$f(x,y) \bigstar h(x,y) = \sum_{m=0}^{M} \sum_{n=0}^{N} f(m,n) h(x-m, y-n)$$

2D discrete convolution theorem

$$f(x,y) \bigstar h(x,y) = \sum_{m=0}^{M} \sum_{n=0}^{N} f(m,n) h(x-m, y-n)$$

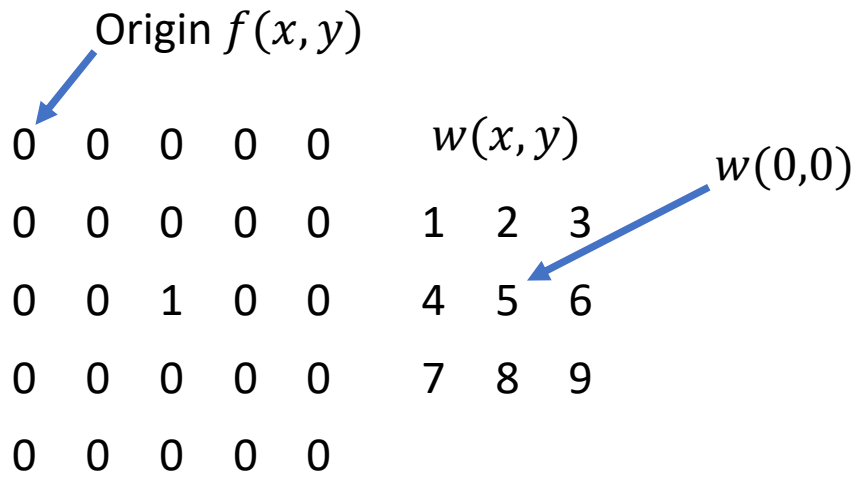
Multiplication in time leads to convolution in frequency:

$$f(x,y)h(x,y) \Leftrightarrow \frac{1}{MN} F(u,v) \bigstar H(u,v)$$

Convolution in time leads to multiplication in frequency:

$$f(x,y) \bigstar h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

Conv. and Corr. points (Example)



w size is 3×3 ➔ $(3-1)/2=1$ padding row and column at the beginning and at the end

new $f(x,y)$ with new padded rows and columns (red)

Origin

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

Conv. and Corr. points (Example)

Origin $f(x,y)$

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

$w(0,0)$

w size is 3×3 ➔ $(3-1)/2=1$ padding row and column at the beginning and at the end

new $f(x,y)$ with new padded rows and columns (red)

Origin

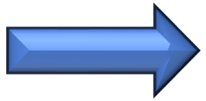
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Rotated $w(x,y)$

9	8	7
6	5	4
3	2	1

Conv. and Corr. points (Example)

correlation



new $f(x,y)$ with new padded rows and columns (red)

Origin

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

Multiply w elements with $f(x,y)$ elements and the origin will be replaced with the multiplication output

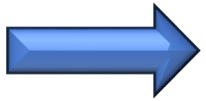
9*0 + 8*0 + … + 1*0 = 0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	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

Step 1: put w on the origin of $f(x,y)$ (kernel elements are in green)

Conv. and Corr. points (Example)

correlation



new $f(x, y)$ with new padded rows and columns (red)

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

Multiply w elements with $f(x, y)$ elements and the origin will be replaced with the multiplication output

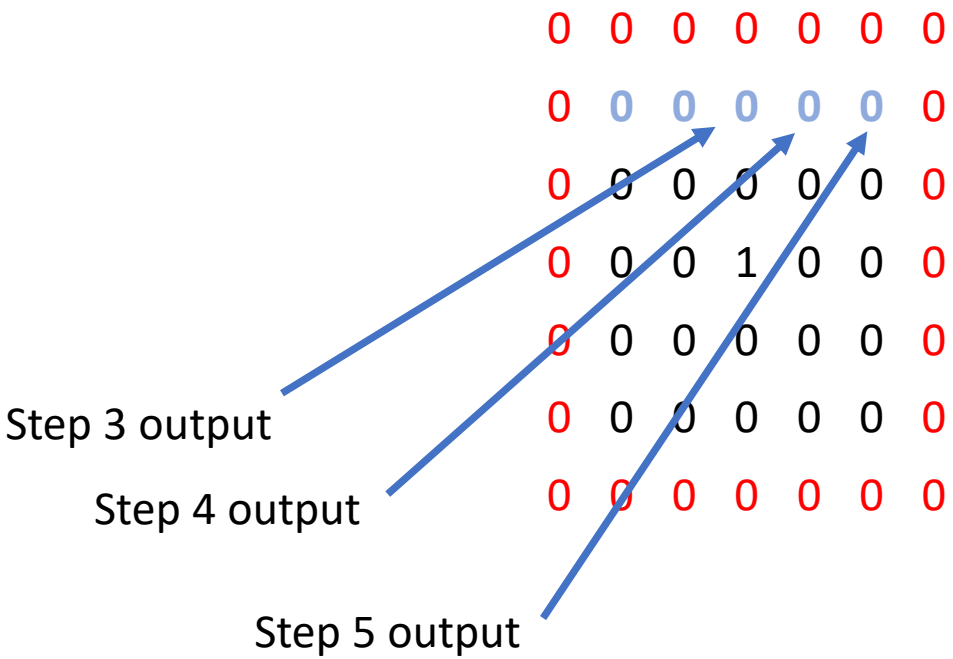
9*0 + 8*0 + … + 1*0 = 0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	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

Step 2: shift kernel w to the right

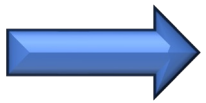
Conv. and Corr. points (Example)

Step 3, 4, and 5: shift kernel w to the right and do the same approach



Conv. and Corr. points (Example)

correlation



new $f(x, y)$ with new padded rows and columns (red)

0	0	0	0	0	0	0
0	9	8	7	0	0	0
0	6	5	4	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

Step 6: shift kernel w to the left and down and do the same approach

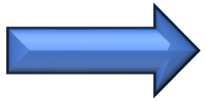
Multiply w elements with $f(x, y)$ elements and the origin will be replaced with the multiplication output

9*0 + 8*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	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

Conv. and Corr. points (Example)

correlation



new $f(x, y)$ with new padded rows and columns (red)

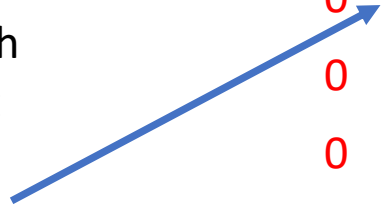
0	0	0	0	0	0	0
0	9	8	7	0	0	0
0	6	5	4	0	0	0
0	3	2	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Step 7: shift kernel w to the right and do the same approach

Multiply w elements with $f(x, y)$ elements and the origin will be replaced with the multiplication output

9*0 + 8*0 + … + 1*1 = 1

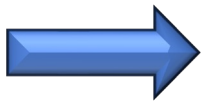
NOTE



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

Conv. and Corr. points (Example)

correlation



new $f(x,y)$ with new padded rows and columns (red)

0	0	0	0	0	0	0
0	0	9	8	7	0	0
0	0	6	5	4	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

Step 8: shift kernel w to the right and do the same approach

Multiply w elements with $f(x,y)$ elements and the origin will be replaced with the multiplication output

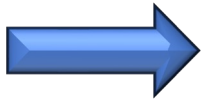
9*0 + …+2*1 + 1*0 = 2

NOTE

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	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

Conv. and Corr. points (Example)

correlation



new $f(x,y)$ with new padded rows and columns (red)

0	0	0	0	0	0	0
0	0	0	9	8	7	0
0	0	0	6	5	4	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

Step 9: shift kernel w to the right and do the same approach

Multiply w elements with $f(x,y)$ elements and the origin will be replaced with the multiplication output

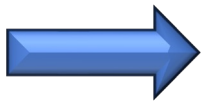
9*0 + …+3*1 + 2*0 + 1*0 = 3

NOTE

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	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

Conv. and Corr. points (Example)

correlation



new $f(x,y)$ with new padded rows and columns (red)

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

Multiply w elements with $f(x,y)$ elements and the origin will be replaced with the multiplication output

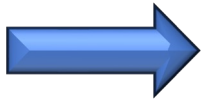
9*0 + … + 2*0 + 1*0 = 0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	0	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

Step 10: shift kernel w to the right and do the same approach

Conv. and Corr. points (Example)

correlation



new $f(x, y)$ with new padded rows and columns (red)

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

9 8 7
6 5 4
3 2 1

Step 11: shift kernel w to the left and down and do the same approach

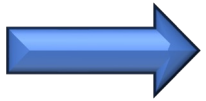
Multiply w elements with $f(x, y)$ elements and the origin will be replaced with the multiplication output

9*0 + 8*0 + … + 1*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	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

Conv. and Corr. points (Example)

correlation



new $f(x,y)$ with new padded rows and columns (red)

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

(9 8 7 / 6 5 4 / 3 2 1)

Step 11: shift kernel w to the right

Multiply w elements with $f(x,y)$ elements and the origin will be replaced with the multiplication output

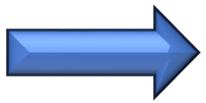
9*0 + … + 4*1 + … + 1*0 = 4

NOTE

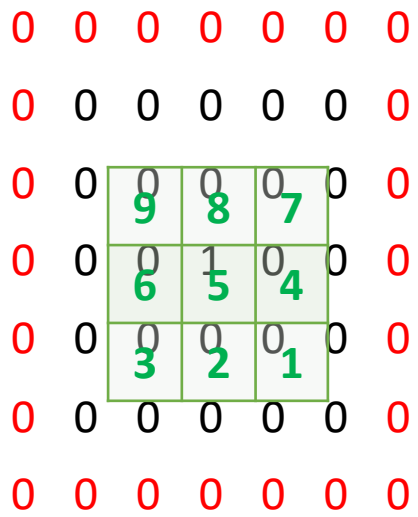
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	4	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Conv. and Corr. points (Example)

correlation



new $f(x, y)$ with new padded rows and columns (red)

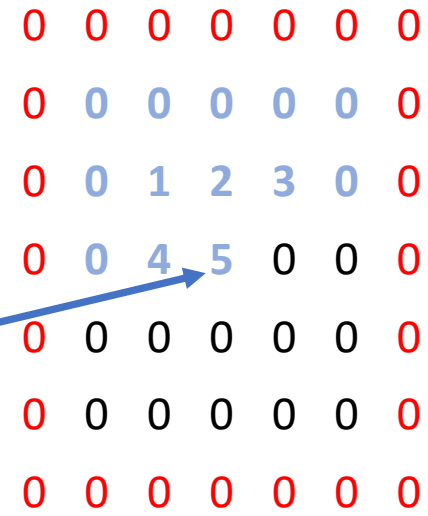


Step 12: shift kernel w to the right

Multiply w elements with $f(x, y)$ elements and the origin will be replaced with the multiplication output

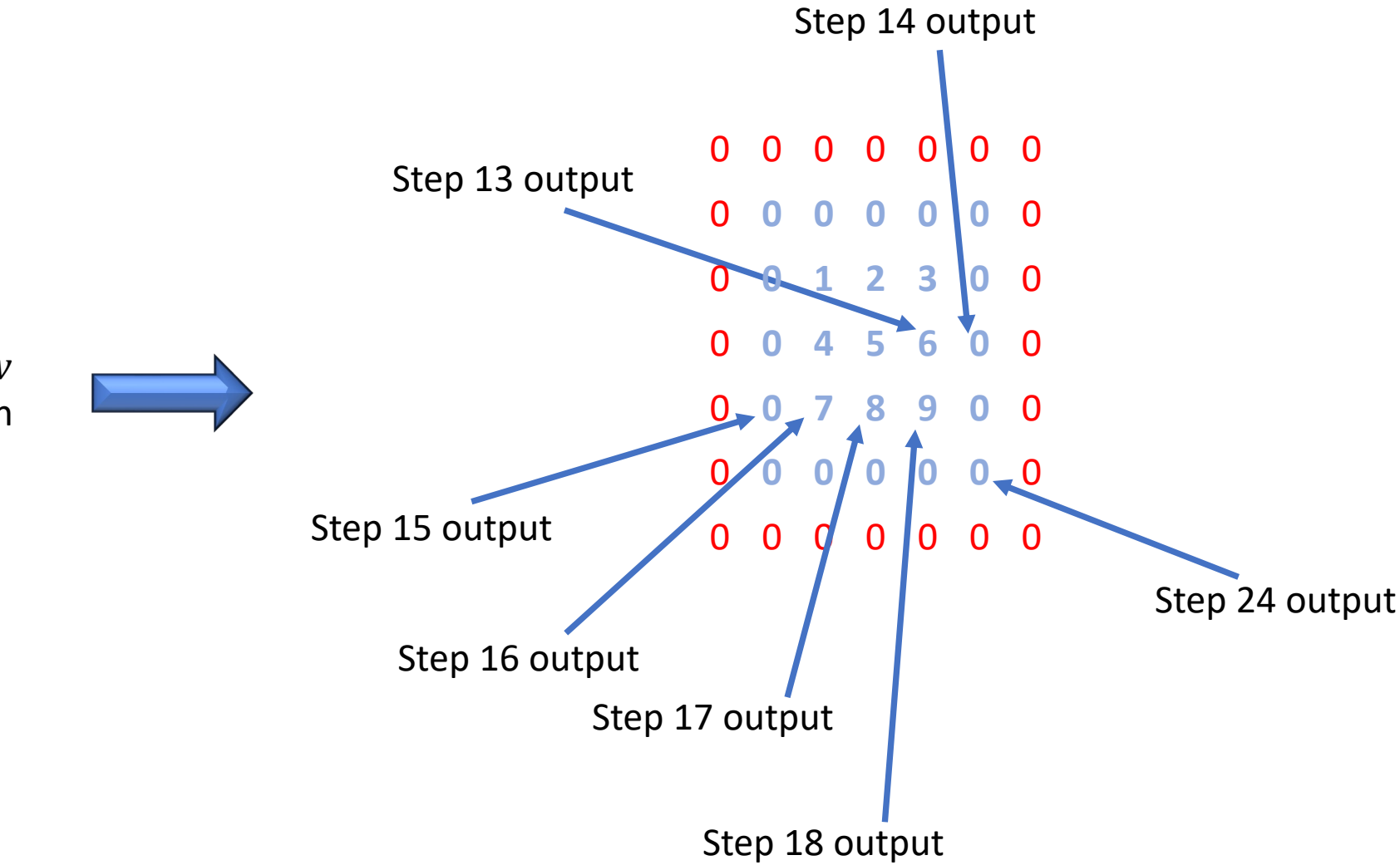
9*0 + … + 5*1 + … + 1*0 = 5

NOTE



Conv. and Corr. points (Example)

Step 13 - 24: shift kernel w and do the same approach

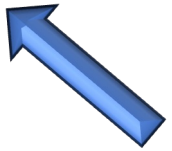


University of Massachusetts Amherst

Conv. and Corr. points (Example)

Full correlation result

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



Correlation 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

Note: This result varies from the correlation result shown on slide 23 of "4_Histogram_v4.pdf." In that case, at the beginning, the first element of the kernel w (not the center of w) aligns with the origin of the image.

Starting from this point onward, please follow the approach demonstrated in this example, where, during the initial step, the center of w aligns with the origin of the image.

2D Discrete Fourier Transform

- 2D Discrete Fourier Transform (DFT)

$$F[k,l] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

where $l = 0, 1, \ldots, N-1$

$k = 0, 1, \ldots, M-1$

- Inverse DFT

$$f[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

2D Discrete Fourier Transform

- It is also possible to define DFT as follows

$$F[k,l] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

where $k = 0,1,\ldots,M-1$

$l = 0,1,\ldots,N-1$

- Inverse DFT

$$f[m,n] = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

2D Discrete Fourier Transform

- Or, as follows

$$F[k,l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

where $k = 0,1,\ldots,M-1$ and $l = 0,1,\ldots,N-1$

- Inverse DFT

$$f[m,n] = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

2D Discrete Fourier Transform

- The discrete two-dimensional Fourier transform of an image array is defined in series form as

$$F(k,l) = \frac{1}{N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi(\frac{k}{M}m+\frac{l}{N}n)}$$

- inverse transform

$$f(m,n) = \frac{1}{N}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} F(k,l)e^{j2\pi(\frac{k}{M}m+\frac{l}{N}n)}$$

Convolution or DFT?

The convolution theorem is your friend!

$$\mathrm{DFT}(f * g) = \mathrm{DFT}(f) \cdot \mathrm{DFT}(\mathrm{g})$$

Convolution in spatial domain is equivalent to **multiplication** in frequency domain!

Filtering with DFT can be **much faster** than image filtering.

Convolution or 2D DFT?

A general linear convolution of $N_1 x N_1$ image with $N_2 x N_2$ convolving function (e.g. smoothing filter) requires in the **image domain** of order

$$N_1^2 N_2^2 \text{ operations.}$$

Instead using **DFT, multiplication, inverse DFT** one needs of order

$$4N^2 \text{Log}_2 N \text{ operations.}$$

Here N is the smallest 2^n number greater or equal to N_1+N_2-1.

Conclusion: Use Image convolution for **small** convolving functions, and DFT for **large** convolving functions.

Convolution or 2D DFT: Example

Example 1: 10x10 pixel image, 5x5 averaging filter

Image domain:
Num. of operations = $10^2 \times 5^2 = 2500$

Using DFT: $N_1+N_2 -1=14$. Smallest 2^n is $2^4=16$. Thus, Num. of operations = $4 \times 16^2 \times \log_2 16 = 4096$

➔ Use image convolution!

Convolution or 2D DFT: Example

Example 2: 100x100 pixel image, 10x10 averaging filter

Image domain:

Num. of operations = $100^2 \times 10^2 = 10^6$

Using DFT: $N_1+N_2-1=109$. Smallest 2^n is $2^7=128$. Thus, Num. of operations = $4 \times 128^2 \times \log_2 16 = 458752 \approx 5 \times 10^5$

➔ Use DFT convolution!

2D Discrete Fourier Transform: Example

Suppose we have as input a constant image of all 1's, $f(m,n) = 1$

2D Discrete Fourier Transform: Example

Suppose we have as input a constant image of all 1's, $f(m,n) = 1$

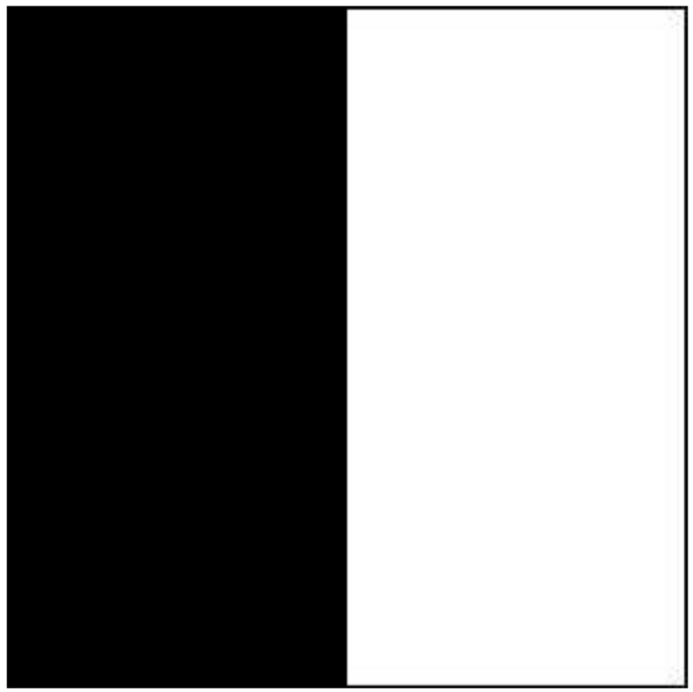
The DFT yields just a DC component, 0 everywhere else.

In this example, DC component is sum of all elements = 64

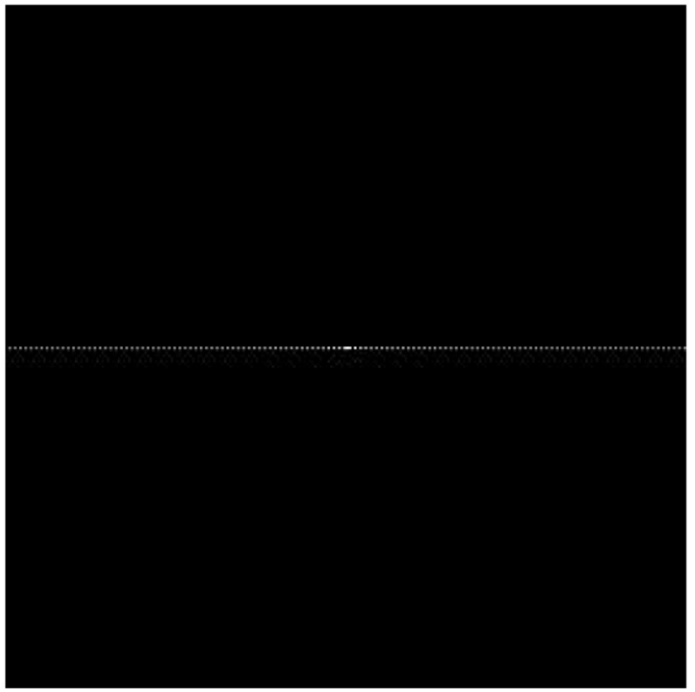
64	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

2D Discrete Fourier Transform: Example

Consider DFT of image with single edge
For display, DC component shifted to center
Log of magnitudes of Fourier Transform displayed



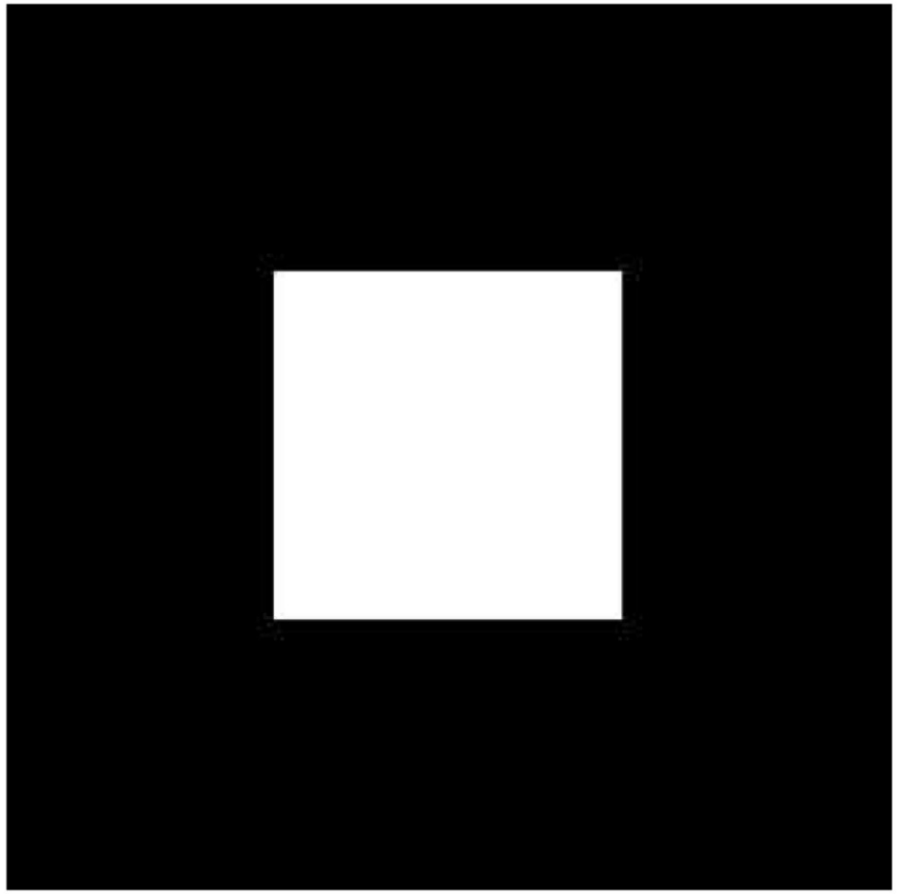
Image



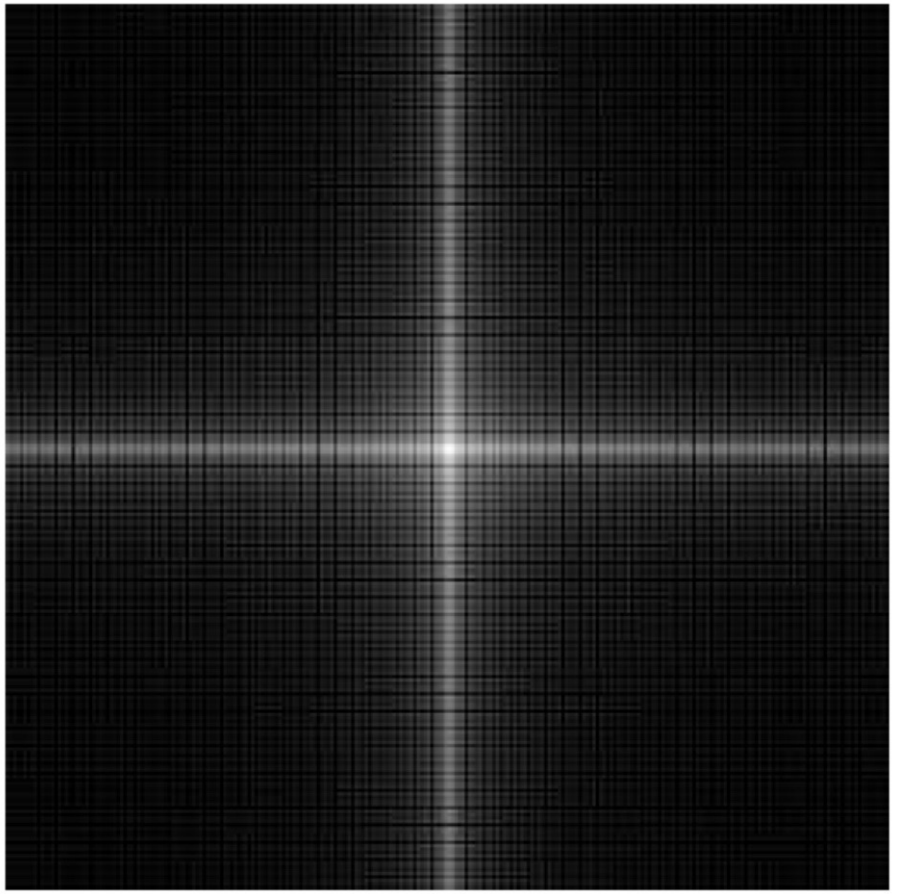
DFT

University of Massachusetts Amherst

2D Discrete Fourier Transform: Example



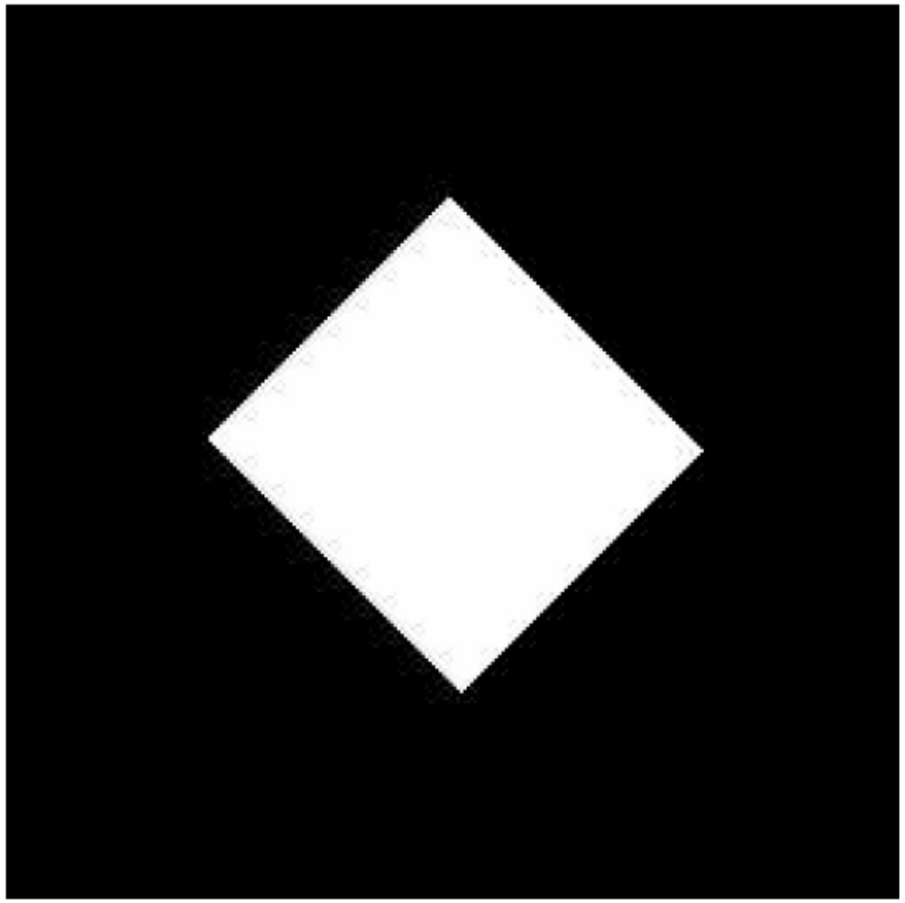
Box



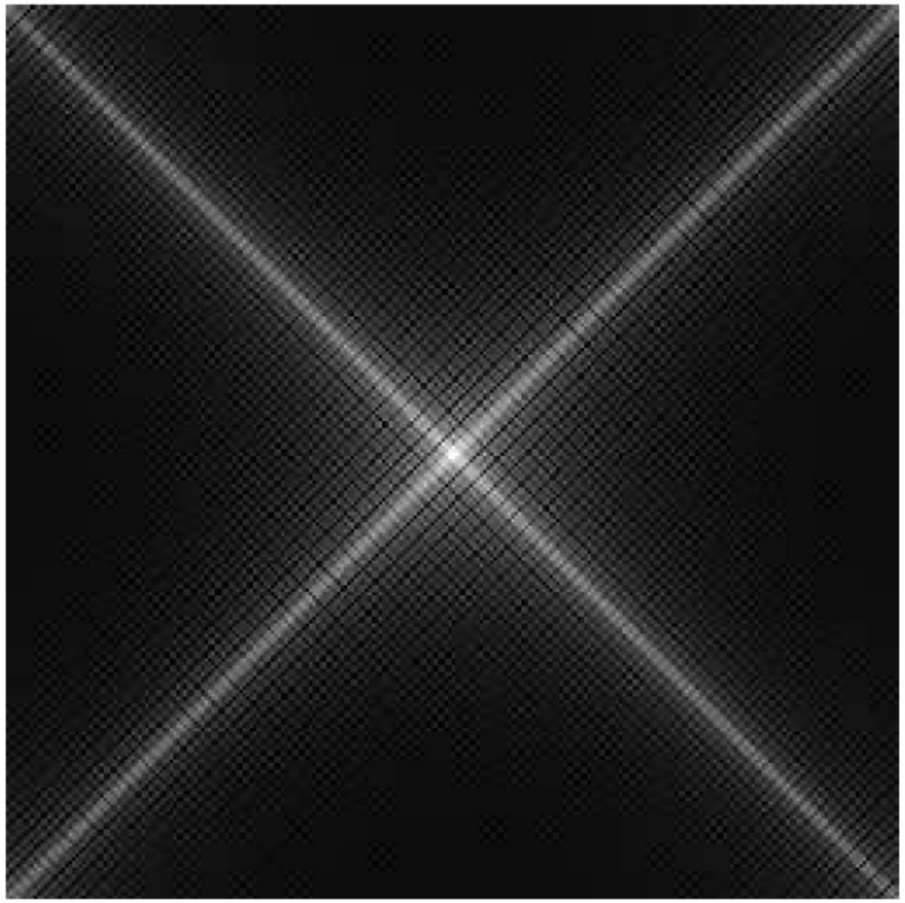
DFT

University of Massachusetts Amherst

2D Discrete Fourier Transform: Example



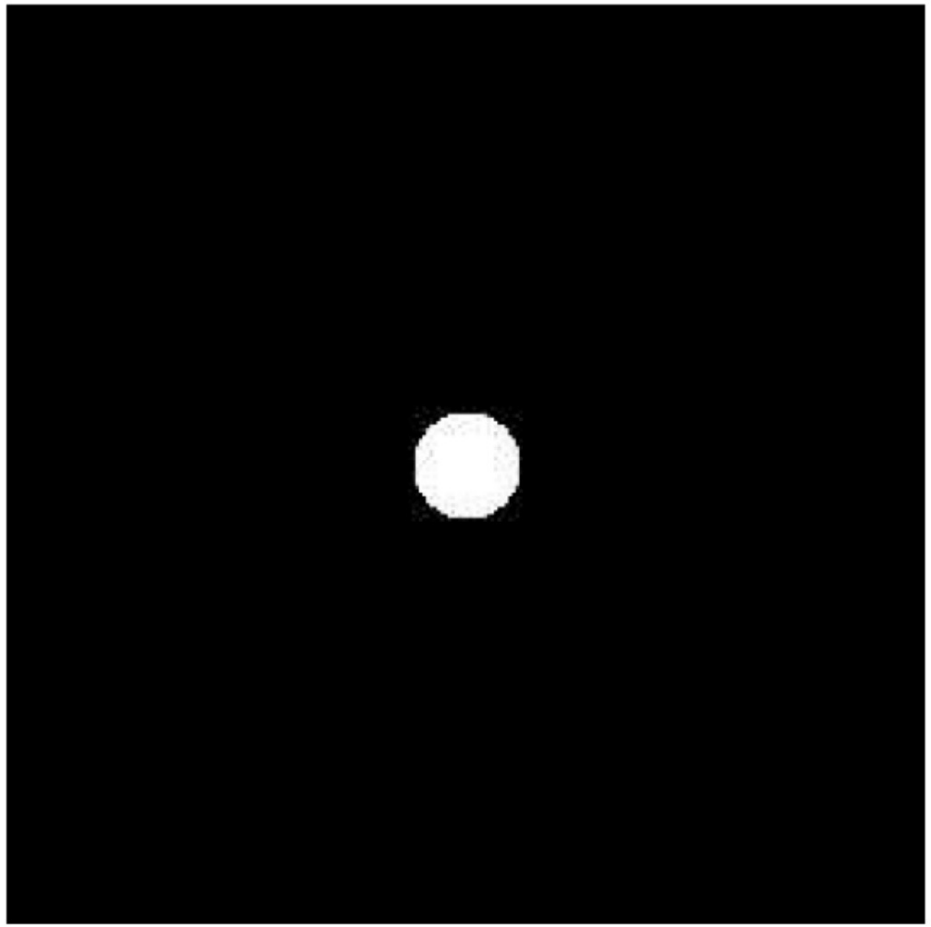
Rotated Box



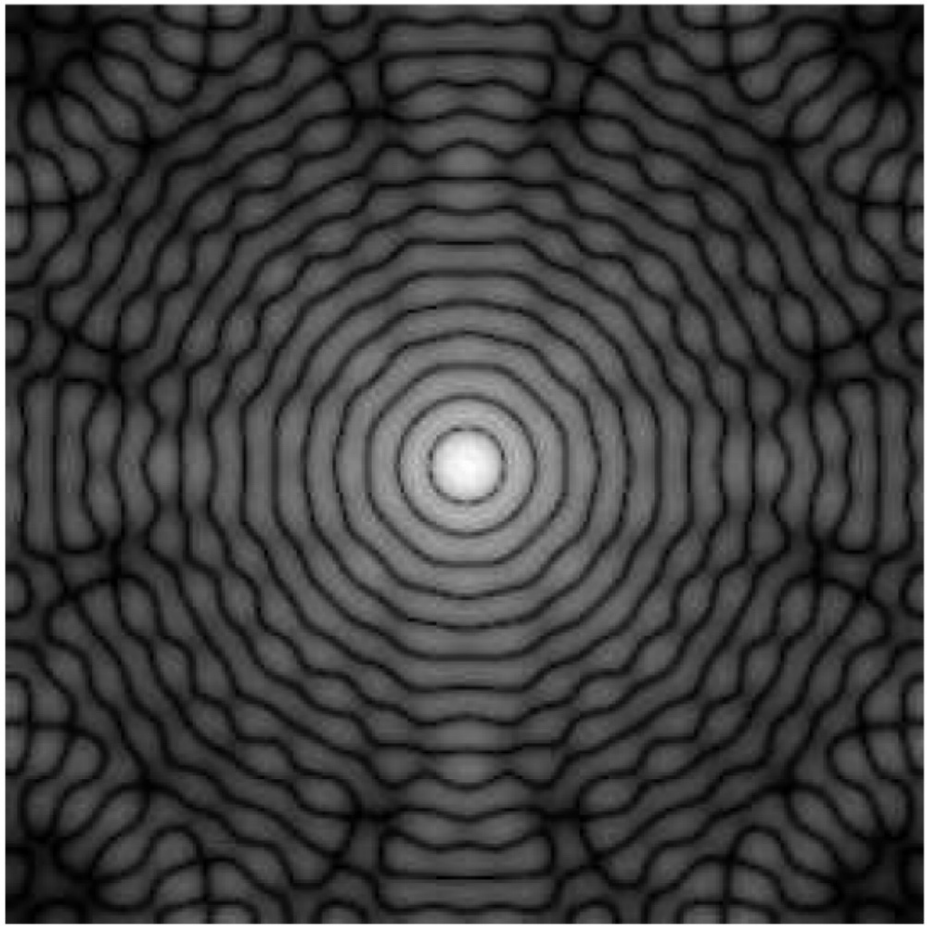
DFT

University of Massachusetts Amherst

2D Discrete Fourier Transform: Example



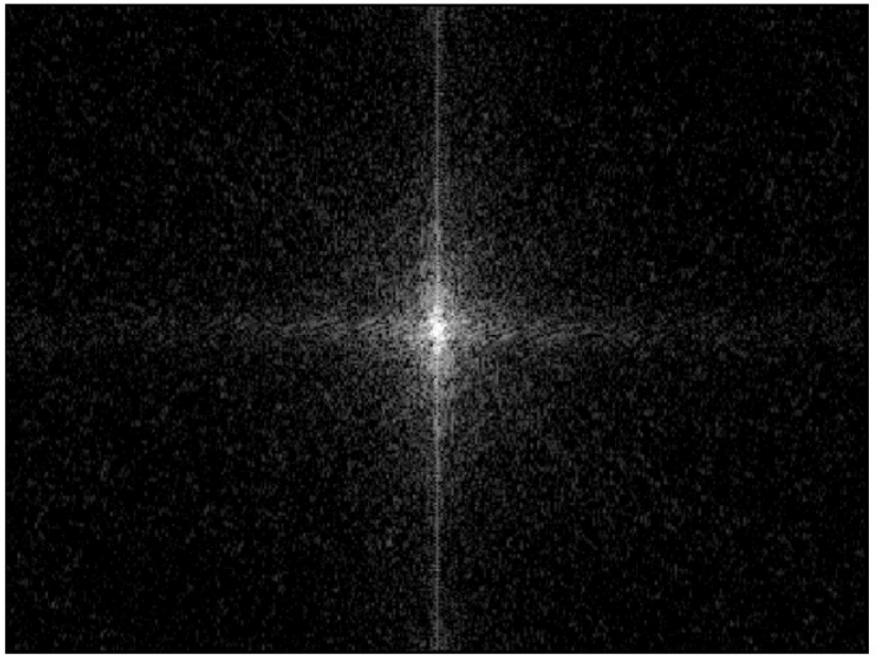
Circle



DFT

University of Massachusetts Amherst

2D Discrete Fourier Transform: Example



University of Massachusetts Amherst

ECE 566 Grading

- ✓ Homework problems: 40%
- ✓ Midterm: 30%
- ✓ Final exam: 30%
- ✓ Online quizzes (Bonus): 5%

University of Massachusetts Amherst

ECE 566 Grading

- ✓ Homework problems: 40%
- ✓ Midterm: 15%
- ✓ Final exam: 15%
- ✓ Online quizzes (Bonus): 5%
- ✓ Final Project! 30%